Учебник для вузов

ПИТЕР

В. Олифер Н. Олифер

Компьютерные сети

Принципы, технологии, протоколы

4-е издание

РЕКОМЕНДОВАНО МИНИСТЕРСТВОМ ОБРАЗОВАНИЯ И НАУКИ РФ

В. Олифер Н. Олифер

Компьютерные сети

Принципы, технологии, протоколы

4-е издание

Рекомендовано Министерством образования и науки Российской Федерации
в качестве учебного пособия для студентов высших учебных заведений,
обучающихся по направлению «Информатика и вычислительная техника»
и по специальностям «Вычислительные машины, комплексы, системы и сети»,
«Автоматизированные машины, комплексы, системы и сети»,
«Программное обеспечение вычислительной техники
и автоматизированных систем».

Москва · Санкт-Петербург · Нижний Новгород · Воронеж
Ростов-на-Дону · Екатеринбург · Самара · Новосибирск
Киев · Харьков Минск
2010

ББК 32.973.202я7
УДК 004.7(075)
О-54

Рецензенты:

Кафедра «Вычислительная техника» факультета «Вычислительные машины и системы» Московского государственного института радиотехники, электроники и автоматики (Технического университета);

Ю. А. Григорьев, д. т. н., профессор кафедры «Системы обработки информации и управления» Московского государственного технического университета им. Н. Э. Баумана;

Б. Ф. Прижуков, к. т. н., заместитель начальника ИВЦ ОАО «Московский междугородный и международный телефон»

Олифер В. Г., Олифер Н. А.

О-54 Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 4-е изд. — СПб.: Питер, 2010. — 944 с.: ил.

ISBN 978-5-49807-389-7

Новое издание одного из лучших российских учебников по сетевым технологиям можно считать юбилейным. Прошло ровно 10 лет с момента первой публикации книги «Компьютерные сети. Принципы, технологии, протоколы». За это время книга приобрела широкую популярность в России, была издана на английском, испанском, португальском и китайском языках, и с каждым новым изданием она существенно обновлялась. Не стало исключением и это, четвертое издание, в котором появилось много новых разделов, посвященных самым актуальным направлениям сетевых технологий.

Издание предназначено для студентов, аспирантов и технических специалистов, которые хотели бы получить базовые знания о принципах построения компьютерных сетей, понять особенности традиционных и перспективных технологий локальных и глобальных сетей, изучить способы создания крупных составных сетей и управления такими сетями.

Рекомендовано Министерством образования и науки Российской Федерации в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлению «Информатика и вычислительная техника» и по специальностям «Вычислительные машины, комплексы, системы и сети», «Автоматизированные машины, комплексы, системы и сети», «Программное обеспечение вычислительной техники и автоматизированных систем».

ББК 32.973.202я7
УДК 004.7(075)

Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-5-49807-389-7

© ООО «Лидер», 2010

Оглавление

От авторов **17**
Для кого эта книга 17
Изменения в четвертом издании 18
Структура книги 19
Веб-сайт поддержки книги 21
Благодарности 22

ЧАСТЬ I. ОСНОВЫ СЕТЕЙ ПЕРЕДАЧИ ДАННЫХ

Глава 1. Эволюция компьютерных сетей **24**
Два корня компьютерных сетей 25
Вычислительная и телекоммуникационная технологии 25
Системы пакетной обработки 25
Многотерминальные системы — прообраз сети 26
Первые компьютерные сети 28
Первые глобальные сети 28
Первые локальные сети 30
Конвергенция сетей 33
Сближение локальных и глобальных сетей 33
Конвергенция компьютерных и телекоммуникационных сетей 35
Выводы 37
Вопросы и задания 37

Глава 2. Общие принципы построения сетей **39**
Простейшая сеть из двух компьютеров 40
Совместное использование ресурсов 40
Сетевые интерфейсы 40
Связь компьютера с периферийным устройством 42
Обмен данными между двумя компьютерами 43
Доступ к ПУ через сеть 44
Сетевое программное обеспечение 44
Сетевые службы и сервисы 45
Сетевая операционная система 47
Сетевые приложения 49
Физическая передача данных по линиям связи 52
Кодирование 52
Характеристики физических каналов 54
Проблемы связи нескольких компьютеров 55
Топология физических связей 55
Адресация узлов сети 59
Коммутация 62

Обобщенная задача коммутации ... 62
Определение информационных потоков ... 63
Маршрутизация ... 64
Продвижение данных ... 67
Мультиплексирование и демультиплексирование ... 69
Разделяемая среда передачи данных ... 70
Типы коммутации ... 73
Выводы ... 74
Вопросы и задания ... 75

Глава 3. Коммутация каналов и пакетов ... 77

Коммутация каналов ... 78
Элементарный канал ... 78
Составной канал ... 80
Неэффективность при передаче пульсирующего трафика ... 84
Коммутация пакетов ... 85
Буферизация пакетов ... 88
Дейтаграммная передача ... 89
Передача с установлением логического соединения ... 91
Передача с установлением виртуального канала ... 93
Сравнение сетей с коммутацией пакетов и каналов ... 95
Транспортная аналогия для сетей с коммутацией пакетов и каналов ... 95
Количественное сравнение задержек ... 96
Ethernet — пример стандартной технологии с коммутацией пакетов ... 103
Выводы ... 105
Вопросы и задания ... 106

Глава 4. Архитектура и стандартизация сетей ... 108

Декомпозиция задачи сетевого взаимодействия ... 109
Многоуровневый подход ... 109
Протокол и стек протоколов ... 112
Модель OSI ... 113
Общая характеристика модели OSI ... 113
Физический уровень ... 116
Канальный уровень ... 116
Сетевой уровень ... 118
Транспортный уровень ... 121
Сеансовый уровень ... 122
Уровень представления ... 122
Прикладной уровень ... 123
Модель OSI и сети с коммутацией каналов ... 123
Стандартизация сетей ... 124
Понятие открытой системы ... 124
Источники стандартов ... 125
Стандартизация Интернета ... 126
Стандартные стеки коммуникационных протоколов ... 126
Соответствие популярных стеков протоколов модели OSI ... 130
Информационные и транспортные услуги ... 131
Распределение протоколов по элементам сети ... 132
Вспомогательные протоколы транспортной системы ... 134
Выводы ... 136
Вопросы и задания ... 136

Глава 5. Примеры сетей ... 138

Классификация компьютерных сетей ... 139
Классификация компьютерных сетей в технологическом аспекте ... 139
Другие аспекты классификации компьютерных сетей ... 141

Обобщенная структура телекоммуникационной сети ... 143
Сеть доступа ... 143
Магистральная сеть ... 144
Информационные центры ... 144
Сети операторов связи ... 145
Услуги ... 146
Клиенты ... 147
Инфраструктура ... 148
Территория покрытия ... 149
Взаимоотношения между операторами связи различного типа ... 150
Корпоративные сети ... 151
Сети отделов ... 151
Сети зданий и кампусов ... 153
Сети масштаба предприятия ... 154
Интернет ... 156
Уникальность Интернета ... 157
Структура Интернета ... 158
Классификация провайдеров Интернета по видам оказываемых услуг ... 159
Выводы ... 160
Вопросы и задания ... 160

Глава 6. Сетевые характеристики ... 162

Типы характеристик ... 163
Субъективные оценки качества ... 163
Характеристики и требования к сети ... 163
Временная шкала ... 164
Соглашение об уровне обслуживания ... 165
Производительность ... 165
Идеальная сеть ... 165
Статистические оценки характеристик сети ... 168
Активные и пассивные измерения в сети ... 171
Характеристики задержек пакетов ... 174
Характеристики скорости передачи ... 177
Надежность ... 179
Характеристики потерь пакетов ... 179
Доступность и отказоустойчивость ... 179
Характеристики сети поставщика услуг ... 180
Расширяемость и масштабируемость ... 180
Управляемость ... 181
Совместимость ... 182
Выводы ... 182
Вопросы и задания ... 183

Глава 7. Методы обеспечения качества обслуживания ... 184

Обзор методов обеспечения качества обслуживания ... 185
Приложения и качество обслуживания ... 187
Предсказуемость скорости передачи данных ... 187
Чувствительность трафика к задержкам пакетов ... 188
Чувствительность трафика к потерям и искажениям пакетов ... 189
Классы приложений ... 190
Анализ очередей ... 191
Модель M/M/1 ... 191
Очереди и различные классы трафика ... 195
Техника управления очередями ... 197
Очередь FIFO ... 197
Приоритетное обслуживание ... 197
Взвешенные очереди ... 200
Комбинированные алгоритмы обслуживания очередей ... 202

Механизмы кондиционирования трафика ... 202
Классификация трафика ... 203
Профилирование ... 203
Формирование трафика ... 204
Обратная связь ... 205
Назначение ... 205
Участники обратной связи ... 206
Информация обратной связи ... 208
Резервирование ресурсов ... 209
Резервирование ресурсов и контроль допуска ... 209
Обеспечение заданного уровня задержек ... 214
Инжиниринг трафика ... 215
Недостатки традиционных методов маршрутизации ... 216
Методы инжиниринга трафика ... 217
Инжиниринг трафика различных классов ... 220
Работа в недогруженном режиме ... 221
Выводы ... 223
Вопросы и задания ... 224

ЧАСТЬ II. ТЕХНОЛОГИИ ФИЗИЧЕСКОГО УРОВНЯ

Глава 8. Линии связи ... 228

Классификация линий связи ... 229
Первичные сети, линии и каналы связи ... 229
Физическая среда передачи данных ... 230
Аппаратура передачи данных ... 232
Характеристики линий связи ... 233
Спектральный анализ сигналов на линиях связи ... 233
Затухание и волновое сопротивление ... 235
Помехоустойчивость и достоверность ... 239
Полоса пропускания и пропускная способность ... 242
Биты и боды ... 244
Соотношение полосы пропускания и пропускной способности ... 246
Типы кабелей ... 247
Экранированная и неэкранированная витая пара ... 247
Коаксиальный кабель ... 249
Волоконно-оптический кабель ... 250
Структурированная кабельная система зданий ... 252
Выводы ... 253
Вопросы и задания ... 254

Глава 9. Кодирование и мультиплексирование данных ... 256

Модуляция ... 257
Модуляция при передаче аналоговых сигналов ... 257
Модуляция при передаче дискретных сигналов ... 257
Комбинированные методы модуляции ... 259
Дискретизация аналоговых сигналов ... 261
Методы кодирования ... 263
Выбор способа кодирования ... 263
Потенциальный код NRZ ... 264
Биполярное кодирование AMI ... 266
Потенциальный код NRZI ... 266
Биполярный импульсный код ... 267
Манчестерский код ... 267
Потенциальный код 2B1Q ... 268

Избыточный код 4B/5B ... 268
Скремблирование ... 269
Компрессия данных ... 272
Обнаружение и коррекция ошибок ... 274
Методы обнаружения ошибок ... 274
Методы коррекции ошибок ... 275
Мультиплексирование и коммутация ... 276
Коммутация каналов на основе методов FDM и WDM ... 276
Коммутация каналов на основе метода TDM ... 278
Дуплексный режим работы канала ... 280
Выводы ... 281
Вопросы и задания ... 282

Глава 10. Беспроводная передача данных ... 284

Беспроводная среда передачи ... 285
Преимущества беспроводных коммуникаций ... 285
Беспроводная линия связи ... 286
Диапазоны электромагнитного спектра ... 287
Распространение электромагнитных волн ... 288
Лицензирование ... 290
Беспроводные системы ... 292
Двухточечная связь ... 292
Связь одного источника и нескольких приемников ... 293
Связь нескольких источников и нескольких приемников ... 295
Типы спутниковых систем ... 296
Геостационарный спутник ... 298
Средне- и низкоорбитальные спутники ... 300
Технология широкополосного сигнала ... 302
Расширение спектра скачкообразной перестройкой частоты ... 302
Прямое последовательное расширение спектра ... 305
Множественный доступ с кодовым разделением ... 306
Выводы ... 308
Вопросы и задания ... 308

Глава 11. Первичные сети ... 310

Сети PDH ... 311
Иерархия скоростей ... 311
Методы мультиплексирования ... 312
Синхронизация сетей PDH ... 314
Ограничения технологии PDH ... 315
Сети SONET/SDH ... 316
Иерархия скоростей и методы мультиплексирования ... 317
Типы оборудования ... 319
Стек протоколов ... 320
Кадры STM-N ... 322
Типовые топологии ... 324
Методы обеспечения живучести сети ... 325
Новое поколение протоколов SDH ... 331
Сети DWDM ... 333
Принципы работы ... 334
Волоконно-оптические усилители ... 335
Типовые топологии ... 336
Оптические мультиплексоры ввода-вывода ... 339
Оптические кросс-коннекторы ... 340

Сети OTN . . . 341
Причины и цели создания . . . 341
Иерархия скоростей . . . 342
Стек протоколов OTN . . . 343
Кадр OTN . . . 344
Выравнивание скоростей . . . 345
Мультиплексирование блоков . . . 346
Коррекция ошибок . . . 346
Выводы . . . 347
Вопросы и задания . . . 348

ЧАСТЬ III. ЛОКАЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

Глава 12. Технологии локальных сетей на разделяемой среде . . . 353

Общая характеристика протоколов локальных сетей на разделяемой среде . . . 354
Стандартная топология и разделяемая среда . . . 354
Стандартизация протоколов локальных сетей . . . 356
Ethernet со скоростью 10 Мбит/с на разделяемой среде . . . 360
MAC-адреса . . . 360
Форматы кадров технологии Ethernet . . . 361
Доступ к среде и передача данных . . . 362
Возникновение коллизии . . . 364
Время оборота и распознавание коллизий . . . 365
Спецификации физической среды . . . 366
Максимальная производительность сети Ethernet . . . 370
Технологии Token Ring и FDDI . . . 372
Беспроводные локальные сети IEEE 802.11 . . . 375
Проблемы и области применения беспроводных локальных сетей . . . 375
Топологии локальных сетей стандарта 802.11 . . . 378
Стек протоколов IEEE 802.11 . . . 380
Распределенный режим доступа DCF . . . 380
Централизованный режим доступа PCF . . . 383
Безопасность . . . 384
Физические уровни стандарта 802.11 . . . 385
Персональные сети и технология Bluetooth . . . 389
Особенности персональных сетей . . . 389
Архитектура Bluetooth . . . 390
Стек протоколов Bluetooth . . . 392
Кадры Bluetooth . . . 394
Поиск и стыковка устройств Bluetooth . . . 395
Пример обмена данными в пикосети . . . 396
Новые свойства Bluetooth . . . 398
Выводы . . . 398
Вопросы и задания . . . 399

Глава 13. Коммутируемые сети Ethernet . . . 402

Мост как предшественник и функциональный аналог коммутатора . . . 403
Логическая структуризация сетей и мосты . . . 403
Алгоритм прозрачного моста IEEE 802.1D . . . 407
Топологические ограничения при применении мостов в локальных сетях . . . 411
Коммутаторы . . . 413
Параллельная коммутация . . . 413
Дуплексный режим работы . . . 417
Неблокирующие коммутаторы . . . 419

Борьба с перегрузками ... 420
Характеристики производительности коммутаторов ... 424
Скоростные версии Ethernet ... 426
Fast Ethernet ... 427
Gigabit Ethernet ... 431
10G Ethernet ... 436
Архитектура коммутаторов ... 438
Конструктивное исполнение коммутаторов ... 442
Выводы ... 445
Вопросы и задания ... 446

Глава 14. Интеллектуальные функции коммутаторов ... 448

Алгоритм покрывающего дерева ... 449
Классическая версия STP ... 449
Версия RSTP ... 456
Агрегирование линий связи в локальных сетях ... 459
Транки и логические каналы ... 459
Борьба с «размножением» пакетов ... 460
Выбор порта ... 462
Фильтрация трафика ... 464
Виртуальные локальные сети ... 467
Назначение виртуальных сетей ... 468
Создание виртуальных сетей на базе одного коммутатора ... 469
Создание виртуальных сетей на базе нескольких коммутаторов ... 470
Альтернативные маршруты в виртуальных локальных сетях ... 474
Качество обслуживания в виртуальных сетях ... 475
Ограничения коммутаторов ... 478
Выводы ... 479
Вопросы и задания ... 479

ЧАСТЬ IV. СЕТИ TCP/IP

Глава 15. Адресация в стеке протоколов TCP/IP ... 482

Стек протоколов TCP/IP ... 483
Типы адресов стека TCP/IP ... 486
Локальные адреса ... 486
Сетевые IP-адреса ... 487
Доменные имена ... 488
Формат IP-адреса ... 488
Классы IP-адресов ... 489
Особые IP-адреса ... 490
Использование масок при IP-адресации ... 492
Порядок назначения IP-адресов ... 493
Назначение адресов автономной сети ... 493
Централизованное распределение адресов ... 494
Адресация и технология CIDR ... 494
Отображение IP-адресов на локальные адреса ... 496
Протокол разрешения адресов ... 497
Протокол Proxy-ARP ... 501
Система DNS ... 502
Плоские символьные имена ... 502
Иерархические символьные имена ... 503
Схема работы DNS ... 505
Обратная зона ... 507

Протокол DHCP . . . 508
Режимы DHCP . . . 508
Алгоритм динамического назначения адресов . . . 510
Выводы . . . 512
Вопросы и задания . . . 512

Глава 16. Протокол межсетевого взаимодействия . . . 514

Формат IP-пакета . . . 515
Схема IP-маршрутизации . . . 517
Упрощенная таблица маршрутизации . . . 519
Таблицы маршрутизации конечных узлов . . . 521
Просмотр таблиц маршрутизации без масок . . . 522
Примеры таблиц маршрутизации разных форматов . . . 523
Источники и типы записей в таблице маршрутизации . . . 527
Пример IP-маршрутизации без масок . . . 528
Маршрутизация с использованием масок . . . 533
Структуризация сети масками одинаковой длины . . . 534
Просмотр таблиц маршрутизации с учетом масок . . . 536
Использование масок переменной длины . . . 538
Перекрытие адресных пространств . . . 541
CIDR . . . 544
Фрагментация IP-пакетов . . . 547
Параметры фрагментации . . . 547
Механизм фрагментации . . . 548
Выводы . . . 550
Вопросы и задания . . . 551

Глава 17. Базовые протоколы TCP/IP . . . 553

Протоколы транспортного уровня TCP и UDP . . . 554
Порты и сокеты . . . 554
Протокол UDP и UDP-дейтаграммы . . . 557
Протокол TCP и TCP-сегменты . . . 558
Логические соединения — основа надежности TCP . . . 560
Повторная передача и скользящее окно . . . 564
Реализация метода скользящего окна в протоколе TCP . . . 567
Управление потоком . . . 570
Общие свойства и классификация протоколов маршрутизации . . . 572
Протокол RIP . . . 575
Построение таблицы маршрутизации . . . 575
Адаптация маршрутизаторов RIP к изменениям состояния сети . . . 578
Пример зацикливания пакетов . . . 580
Методы борьбы с ложными маршрутами в протоколе RIP . . . 581
Протокол OSPF . . . 582
Два этапа построения таблицы маршрутизации . . . 583
Метрики . . . 584
Маршрутизация в неоднородных сетях . . . 585
Взаимодействие протоколов маршрутизации . . . 585
Внутренние и внешние шлюзовые протоколы . . . 586
Протокол BGP . . . 588
Протокол ICMP . . . 591
Утилита traceroute . . . 593
Утилита ping . . . 596
Выводы . . . 597
Вопросы и задания . . . 598

Глава 18. Дополнительные функции маршрутизаторов IP-сетей ... 599
Фильтрация ... 600
Фильтрация пользовательского трафика ... 600
Фильтрация маршрутных объявлений ... 603
Стандарты QoS в IP-сетях ... 603
Модели качества обслуживания IntServ и DiffServ ... 604
Алгоритм ведра маркеров ... 605
Случайное раннее обнаружение ... 607
Интегрированное обслуживание и протокол RSVP ... 608
Дифференцированное обслуживание ... 611
Трансляция сетевых адресов ... 616
Причины подмены адресов ... 616
Традиционная технология NAT ... 616
Базовая трансляция сетевых адресов ... 618
Трансляция сетевых адресов и портов ... 619
Групповое вещание ... 621
Стандартная модель группового вещания IP ... 622
Адреса группового вещания ... 626
Основные типы протоколов группового вещания ... 627
Протокол IGMP ... 627
Принципы маршрутизации трафика группового вещания ... 630
Протокол DVMRP ... 632
Протокол MOSPF ... 635
Протокол PIM-SM ... 635
IPv6 как развитие стека TCP/IP ... 640
Система адресации протокола IPv6 ... 641
Снижение нагрузки на маршрутизаторы ... 644
Переход на версию IPv6 ... 647
Маршрутизаторы ... 648
Функции маршрутизаторов ... 648
Классификация маршрутизаторов по областям применения ... 651
Выводы ... 657
Вопросы и задания ... 657

ЧАСТЬ V. ТЕХНОЛОГИИ ГЛОБАЛЬНЫХ СЕТЕЙ

Глава 19. Транспортные услуги и технологии глобальных сетей ... 661
Базовые понятия ... 662
Типы публичных услуг сетей операторов связи ... 662
Многослойная сеть оператора связи ... 666
Технология Frame Relay ... 672
История стандарта ... 672
Техника продвижения кадров ... 672
Гарантии пропускной способности ... 675
Технология ATM ... 678
Ячейки ATM ... 678
Виртуальные каналы ATM ... 680
Категории услуг ATM ... 681
Виртуальные частные сети ... 682
IP в глобальных сетях ... 689
Чистая IP-сеть ... 689
Протокол HDLC ... 690
Протокол PPP ... 690
Использование выделенных линий IP-маршрутизаторами ... 692
Работа IP-сети поверх сети ATM ... 693

Выводы . . . 695
Вопросы и задания . . . 696

Глава 20. Технология MPLS . . . 698

Базовые принципы и механизмы MPLS . . . 699
Совмещение коммутации и маршрутизации в одном устройстве . . . 699
Пути коммутации по меткам . . . 702
Заголовок MPLS и технологии канального уровня . . . 704
Стек меток . . . 706
Протокол LDP . . . 709
Мониторинг состояния путей LSP . . . 714
Тестирование путей LSP . . . 715
Трассировка путей LSP . . . 716
Протокол двунаправленного обнаружения ошибок продвижения . . . 717
Инжиниринг трафика в MPLS . . . 717
Отказоустойчивость путей MPLS . . . 722
Общая характеристика . . . 722
Использование иерархии меток для быстрой защиты . . . 723
Выводы . . . 724
Вопросы и задания . . . 725

Глава 21. Ethernet операторского класса . . . 727

Обзор версий Ethernet операторского класса . . . 728
Движущие силы экспансии Ethernet . . . 728
Разные «лица» Ethernet . . . 729
Стандартизация Ethernet как услуги . . . 731
Технология EoMPLS . . . 733
Псевдоканалы . . . 733
Услуги VPWS . . . 737
Услуги VPLS . . . 739
Ethernet поверх Ethernet . . . 741
Области улучшений Ethernet . . . 741
Функции эксплуатации, администрирования и обслуживания в Ethernet . . . 743
Мосты провайдера . . . 746
Магистральные мосты провайдера . . . 748
Магистральные мосты провайдера с поддержкой инжиниринга трафика . . . 753
Выводы . . . 756
Вопросы и задания . . . 757

Глава 22. Удаленный доступ . . . 759

Схемы удаленного доступа . . . 760
Типы клиентов и абонентских окончаний . . . 761
Мультиплексирование информации на абонентском окончании . . . 763
Режим удаленного узла . . . 765
Режим удаленного управления и протокол telnet . . . 767
Коммутируемый аналоговый доступ . . . 768
Принцип работы телефонной сети . . . 769
Удаленный доступ через телефонную сеть . . . 771
Модемы . . . 772
Коммутируемый доступ через сеть ISDN . . . 775
Назначение и структура ISDN . . . 775
Интерфейсы BRI и PRI . . . 777
Стек протоколов ISDN . . . 779
Использование сети ISDN для передачи данных . . . 781
Технология ADSL . . . 784

Доступ через сети CATV . . . 787
Беспроводной доступ . . . 789
Выводы . . . 791
Вопросы и задания . . . 792

Глава 23. Сетевые службы . . . 794

Электронная почта . . . 795
Электронные сообщения . . . 795
Протокол SMTP . . . 795
Непосредственное взаимодействие клиента и сервера . . . 796
Схема с выделенным почтовым сервером . . . 797
Схема с двумя почтовыми серверами-посредниками . . . 799
Протоколы POP3 и IMAP . . . 800
Веб-служба . . . 801
Веб- и HTML-страницы . . . 802
URL . . . 803
Веб-клиент и веб-сервер . . . 803
Протокол HTTP . . . 805
Формат HTTP-сообщений . . . 806
Динамические веб-страницы . . . 807
IP-телефония . . . 808
Ранняя IP-телефония . . . 808
Стандарты H.323 . . . 809
Стандарты на основе протокола SIP . . . 811
Связь телефонных сетей через Интернет . . . 813
Новое поколение сетей IP-телефонии . . . 814
Распределенные шлюзы и программные коммутаторы . . . 816
Новые услуги . . . 817
Интеграция систем адресации E.164 и DNS на основе ENUM . . . 818
Протокол передачи файлов . . . 819
Основные модули службы FTP . . . 819
Управляющий сеанс и сеанс передачи данных . . . 820
Команды взаимодействия FTP-клиента с FTP-сервером . . . 820
Сетевое управление в IP-сетях . . . 821
Функции систем управления . . . 821
Архитектуры систем управления сетями . . . 823
Выводы . . . 826
Вопросы и задания . . . 827

Глава 24. Сетевая безопасность . . . 828

Основные понятия информационной безопасности . . . 829
Определение безопасной системы . . . 829
Угроза, атака, риск . . . 830
Типы и примеры атак . . . 831
Атаки отказа в обслуживании . . . 831
Перехват и перенаправление трафика . . . 833
Внедрение в компьютеры вредоносных программ . . . 837
Троянские программы . . . 837
Сетевые черви . . . 838
Вирусы . . . 842
Шпионские программы . . . 844
Спам . . . 844
Методы обеспечения информационной безопасности . . . 845
Классификация методов защиты . . . 845
Политика безопасности . . . 846

Шифрование ... 847
Симметричные алгоритмы шифрования ... 848
Алгоритм DES ... 849
Несимметричные алгоритмы шифрования ... 850
Алгоритм RSA ... 853
Односторонние функции шифрования ... 854
Аутентификация, авторизации, аудит ... 856
Понятие аутентификации ... 856
Авторизация доступа ... 858
Аудит ... 859
Строгая аутентификация на основе многоразового пароля в протоколе CHAP ... 860
Аутентификация на основе одноразового пароля ... 861
Аутентификация на основе сертификатов ... 863
Аутентификация информации ... 869
Антивирусная защита ... 871
Сканирование сигнатур ... 872
Метод контроля целостности ... 873
Сканирование подозрительных команд ... 874
Отслеживание поведения программ ... 874
Сетевые экраны ... 875
Типы сетевых экранов разных уровней ... 878
Реализация ... 879
Архитектура ... 880
Прокси-серверы ... 882
Функции прокси-сервера ... 882
Прокси-серверы прикладного уровня и уровня соединений ... 885
«Проксификация» приложений ... 886
Системы обнаружения вторжений ... 887
Протоколы защищенного канала. IPsec ... 887
Иерархия технологий защищенного канала ... 889
Распределение функций между протоколами IPSec ... 890
Безопасная ассоциация ... 891
Транспортный и туннельный режимы ... 893
Протокол AH ... 895
Протокол ESP ... 896
Базы данных SAD И SPD ... 898
Сети VPN на основе шифрования ... 900
Выводы ... 902
Вопросы и задания ... 903

Ответы на вопросы ... 905

Рекомендуемая и использованная литература ... 917

Алфавитный указатель ... 918

Посвящаем нашей дочери Анне

От авторов

Эта книга является результатом многолетнего опыта преподавания авторами курсов сетевой тематики в аудиториях государственных вузов, коммерческих учебных центров, а также учебных центров предприятий и корпораций.

Основу книги составили материалы курсов «Проблемы построения корпоративных сетей», «Основы сетевых технологий», «Организация удаленного доступа», «Сети TCP/IP», «Стратегическое планирование сетей масштаба предприятия» и ряда других. Эти материалы прошли успешную проверку в бескомпромиссной и сложной аудитории, состоящей из слушателей с существенно разным уровнем подготовки и кругом профессиональных интересов. Среди них были студенты и аспиранты вузов, сетевые администраторы и интеграторы, начальники отделов автоматизации и преподаватели. Учитывая специфику аудитории, курсы лекций строились так, чтобы начинающий получил основу для дальнейшего изучения, а специалист систематизировал и актуализировал свои знания. В соответствии с такими же принципами написана и эта книга — она является фундаментальным курсом по компьютерным сетям, который сочетает широту охвата основных областей, проблем и технологий этой быстро развивающейся области знаний с основательным рассмотрением деталей каждой технологии и особенностей оборудования.

Для кого эта книга

Книга предназначена для студентов, аспирантов и технических специалистов, которые хотят получить базовые знания о принципах построения компьютерных сетей, понять особенности традиционных и перспективных технологий локальных и глобальных сетей, изучить способы создания крупных составных сетей и управления такими сетями.

Книга будет полезна начинающим специалистам в области сетевых технологий, которые имеют только общие представления о работе сетей из опыта общения с персональными компьютерами и Интернетом, но хотели бы получить фундаментальные знания, позволяющие продолжить изучение сетей самостоятельно.

Сложившимся сетевым специалистам книга может помочь в знакомстве с теми технологиями, с которыми им не приходилось сталкиваться в практической работе, систематизировать

имеющиеся знания, стать справочником, позволяющим найти описание конкретного протокола, формата кадра и т. п. Кроме того, книга дает необходимую теоретическую основу для подготовки к сертификационным экзаменам таких компаний, как Cisco CCNA, CCNP, CCDP и CCIP.

Студенты высших учебных заведений, обучающиеся по направлению «220000. Информатика и вычислительная техника» и по специальностям «Вычислительные машины, комплексы, системы и сети», «Автоматизированные машины, комплексы, системы и сети», «Программное обеспечение вычислительной техники и автоматизированных систем», могут использовать книгу в качестве рекомендованного Министерством образования Российской Федерации учебного пособия.

Изменения в четвертом издании

Прошло ровно 10 лет со времени публикации первого издания этой книги. И с каждым новым изданием она существенно обновлялась. Не стало исключением и это, четвертое, издание. Одни разделы претерпели значительные изменения, а другие, которые потеряли свою актуальность и стали интересны лишь узкому кругу специалистов, были вовсе исключены из книги и перенесены на веб-сайт поддержки этой книги.

И, конечно, в книге появилось много нового. Так, в книге появилось три новые главы.

- Глава 21, «Ethernet операторского класса». Технология, давшая название этой главе, известная также как Carrier Ethernet, появилась совсем недавно, но ее популярность быстро растет. Выход Ethermet за пределы локальных сетей является знаковым событием, обещающим новые возможности как для пользователей, так и для провайдеров. В этой главе рассматриваются две основные ветви данной технологии: на базе MPLS и на базе усовершенствованной версии Ethernet.
- Глава 23, «Сетевые службы». В ответ на пожелания многих наших читателей мы расширили освещение сетевых средств прикладного уровня, добавив описания таких служб, как электронная почта, WWW и IP-телефония.
- Глава 24, «Сетевая безопасность». Появление этой главы отражает всевозрастающую обеспокоенность интернет-сообщества проблемами информационной защиты. В этой главе приведены описания различных угроз безопасности компьютерных сетей, связанных с внедрением вредоносных программ (вирусов, червей, троянских и шпионских программ), DoS-атаками, ответвлением трафика. Также рассматриваются методы и средства предупреждения и обнаружения атак: шифрование, аутентификация, авторизация, антивирусная защита, сетевые экраны, прокси-серверы, протоколы защищенного канала и виртуальные частные сети на основе шифрования.

Помимо отдельных глав в книге появилось несколько новых разделов.

В главу 7 добавлен раздел «Работа в недогруженном режиме». В нем описывается широко распространенная практика обеспечения временных характеристик передачи пакетов за счет поддержания избыточной пропускной способности.

В главу 11, посвященную первичным сетям, добавлено описание технологии оптических транспортных сетей (OTN), которая обеспечивает мультиплексирование и коммутацию высокоскоростных потоков данных в волновых каналах DWDM. В эту главу включено также описание новых функций технологии SDH, направленных на более эффективную

передачу трафика компьютерных сетей, таких как виртуальная конкатенация (VCAT), схема динамического изменения пропускной способности линии (LCAS) и общая процедура инкапсуляции данных (GFP).

Важным дополнением главы 18 стал раздел «Групповое вещание», освещающий очень перспективное направление в развитии технологии TCP/IP. Групповое вещание лежит в основе бурно развивающихся широковещательных сервисов Интернета, таких как IP-телевидение, аудиовещание, видеоконференции.

Переработанный и дополненный материал о технологии MPLS, которая утвердила себя в качестве надежного фундамента для построения разнообразных транспортных сервисов, выделен в отдельную главу (главу 20).

И наконец, были исправлены мелкие ошибки и опечатки в тексте и рисунках, замеченные читателями и самими авторами.

Структура книги

Книга состоит из 24 глав, объединенных в 5 частей.

В первой части, «Основы сетей передачи данных», состоящей из 7 глав, описаны основные принципы и архитектурные решения, которые лежат в основе всех современных сетевых технологий, рассматриваемых в последующих частях книги. В главе 1, рассказывающей об эволюции компьютерных сетей, особый акцент делается на конвергенции разных видов телекоммуникационных сетей. В главе 2 даются фундаментальные понятия коммутации, мультиплексирования, маршрутизации, адресации и архитектуры сетей. В следующей, третьей, главе обсуждаются два основных подхода к коммутации — коммутация каналов и пакетов. Глава 4 фокусируется на иерархической организации сетей и семиуровневой модели OSI. В главе 5 приводится классификация компьютерных сетей, в ней читатель найдет также описание основных типов сетей: сетей операторов связи, корпоративных сетей и глобальной сети Интернет. Завершают первую часть книги главы 7 и 8, относящиеся к анализу работы сети.

Вторая часть, «Технологии физического уровня», состоит из четырех глав, из которых первые две носят вспомогательный характер. В них описываются различные типы линий связи, детально излагаются современные методы передачи дискретной информации в сетях. Наличие этого материала в учебнике дает возможность читателю, не тратя время на просмотр большого количества литературы, получить необходимый минимум знаний в таких областях, как теория информации, спектральный анализ, физическое и логическое кодирование данных, обнаружение и коррекция ошибок. Глава 10 посвящена беспроводной передаче данных, которая приобретает все большую популярность. Высокий уровень помех и сложные пути распространения волн требуют применения в беспроводных каналах особых способов кодирования и передачи сигналов. В главе 11 изучаются технологии PDH, SDH/SONET, DWDM и OTN, создающие инфраструктуру физических каналов для глобальных телекоммуникационных сетей. На основе каналов, образованных первичными сетями, работают наложенные компьютерные и телефонные сети.

Третья часть, «Локальные вычислительные сети», включает три главы. В главе 12 рассматриваются технологии локальных сетей на разделяемой среде: основное внимание уделено классическим вариантам Ethernet со скоростью 10 Мбит/с на коаксиале и витой

паре; также здесь кратко рассмотрены принципы работы основных соперников Ethernet в 80-е и 90-е годы – технологий Token Ring и FDDI. Приводится описание двух наиболее популярных беспроводных технологий локальных сетей – IEEE 802.11 (LAN) и Bluetooth (PAN). Глава 13 посвящена коммутируемым локальным сетям. В ней рассматриваются основные принципы работы таких сетей: алгоритм функционирования коммутатора локальной сети, дуплексные версии протоколов локальных сетей, особенности реализации коммутаторов локальных сетей. В главе 14 изучаются расширенные возможности коммутируемых локальных сетей этого типа: резервные связи на основе алгоритма покрывающего дерева, агрегирование каналов, а также техника виртуальных локальных сетей, позволяющая быстро и эффективно выполнять логическую структуризацию сети.

Следуя логике, диктуемой моделью OSI, вслед за частями, в которых были рассмотрены технологии физического и канального уровней, *четвертую часть*, «Сети TCP/IP», мы посвящаем средствам сетевого уровня, то есть средствам, которые обеспечивают возможность объединения множества сетей в единую сеть. Учитывая, что бесспорным лидером среди протоколов сетевого уровня является протокол IP, ему в книге уделяется основное внимание. В главе 15 описываются различные аспекты IP-адресации: способы отображения локальных, сетевых и символьных адресов, использование масок и современных методов агрегирования IP-адресов, а также способы автоматического конфигурирования IP-узлов. В главе 16 детально рассмотрена работа протокола IP по продвижению и фрагментации пакетов, изучается общий формат таблицы маршрутизации и примеры ее частных реализаций в программных и аппаратных маршрутизаторах различных типов. При обсуждении особенностей новой версии IPv6 подробно обсуждается схема модернизации адресации, а также изменение формата заголовка IP. Глава 17 начинается с изучения протоколов TCP и UDP, исполняющих посредническую роль между приложениями и транспортной инфраструктурой сети. Далее подробно описываются протоколы маршрутизации RIP, OSPF и BGP, анализируются области применимости этих протоколов и возможности их комбинирования. Завершает главу рассмотрение протокола ICMP, являющегося средством оповещения отправителя о причинах недоставки его пакетов адресату. В главе 18 содержится описание тех функций маршрутизаторов, которые хотя и фигурируют в названии главы как «дополнительные», но без которых трудно представить существование современных компьютерных сетей. К таким функциям относятся трансляция сетевых адресов, фильтрация трафика, поддержка QoS, IPv6 и группового вещания. В завершении этой главы приводится классификация маршрутизаторов на основе их внутренней организации и областей использования. Всестороннее изучение в этой части протоколов стека TCP/IP придает ей самостоятельное значение введения в IP-сети.

Пятая часть, «Технологии глобальных сетей», состоит из шести глав. В главе 19 анализируются три основных типа транспортных услуг, предоставляемых операторами связи: доступ в Интернет, виртуальные частные сети и услуги выделенных каналов. Кроме того, в этой главе рассматривается многоуровневая структура сети оператора связи, включающая уровни первичной сети, канального уровня и уровня IP. Также дается обзор технологий Frame Relay и ATM. Глава 20 посвящена основным принципам и базовым элементам технологии MPLS, таким как протокол LDP, многоуровневая организация соединений, механизмы защиты соединений и тестирования их состояния. В главе 21 описаны различные варианты технологий, объединенных под общим названием Ethernet операторского класса (Carrier Ethernet). В главе 22 рассматриваются схемы и технологии удаленного доступа. Наиболее эффективными являются технологии, в которых используется существующая кабельная инфраструктура (например, линии ADSL, работающие на абонентских окон-

чаниях телефонной сети) или кабельные модемы, опирающиеся на системы кабельного телевидения. Альтернативным решением является беспроводной доступ, как мобильный, так и фиксированный. Прикладные службы глобальных сетей рассматриваются в главе 23. Именно информационные службы, такие как электронная почта и WWW, сделали в свое время Интернет столь популярным. И сегодня популярность Интернета растет благодаря появлению новых сервисов, среди которых в первую очередь нужно отметить IP-телефонию и видеоконференции. Часть, а вместе с ней и книга, завершается главой 24, посвященной сетевой безопасности. Уязвимость Интернета является оборотной стороной его открытости, так как в Интернете каждый может не только общаться с каждым, но и атаковать каждого. Вирусы, черви, распределенные атаки и, наконец, спам — все это, к сожалению, ежедневно мешает «жителям» Интернета нормально жить и работать. В главе 24 анализируются основные типы угроз, присущих глобальным сетям, и изучаются базовые механизмы и технологии защиты от этих угроз.

Авторы стремились сделать работу читателя с книгой максимально эффективной. Подробный индексный указатель позволяет быстро найти интересующий материал по одному из многочисленных терминов, используемых в сетевой индустрии. Каждая глава завершается выводами, которые призваны сконцентрировать внимание читателя на основных идеях, темах и терминах главы, помогая ему не упустить из виду главное за обилием, хотя и полезных, но частных фактов и деталей. В конце каждой главы помещены вопросы и упражнения для проверки степени усвоения основных концепций, а в отдельных случаях и для углубления понимания некоторых идей.

Веб-сайт поддержки книги

Дополнительную информацию по этой и другим книгам авторов читатели могут найти на сайте www.olifer.co.uk. В данный момент на сайте размещены следующие материалы, относящиеся к этому изданию книги:

- Дополнительные разделы, ссылки на которые помещены в тексте книги.
- Все иллюстрации из книги.
- Дополнительные вопросы и задания, а также ответы на них.
- Презентации в форматах Power Point и HTML последовательно по всем главам книги.
- Путеводитель по книге (road map) призван помочь преподавателю при создании учебных курсов на базе этой книги, таких, например, как «Беспроводные системы», «Введение в IP», «Качество обслуживания», «Удаленный доступ» и т. п. В этом путеводителе авторы перечисляют последовательность глав (маршрут), в которых содержится соответствующий материал, и по мере необходимости дают методические советы.
- Дополнительные примеры (case studies) могут быть использованы как темы для курсовых проектов.
- Информационные ресурсы Интернета связаны с темами книги.
- И наконец, мнения, замечания и вопросы читателей, замеченные опечатки и ошибки.

Мы с благодарностью примем ваши отзывы по адресу victor@olifer.co.uk и natalia@olifer.co.uk.

Благодарности

Мы благодарим наших читателей за их многочисленные пожелания, вопросы и замечания.

Мы признательны также всем сотрудникам издательства «Питер», которые принимали участие в создании этой книги. Особая благодарность президенту издательства «Питер» Вадиму Усманову, руководителю проектной группы «Компьютерная литература» Андрею Сандрыкину, ведущему специалисту этой группы Андрею Юрченко и нашему неизменному литературному редактору Алексею Жданову.

Виктор Олифер, к.т.н., ССІР

Наталья Олифер, к.т.н., доцент

От издательства

Подробную информацию о наших книгах вы найдете на веб-сайте издательства www.piter.com. Там же вы можете оставить ваши отзывы и пожелания.

Часть I

Основы сетей передачи данных

Процесс познания всегда развивается по спирали. Мы не можем сразу понять и осознать сложное явление, мы должны рассматривать его с разных точек зрения, в целом и по частям, изолированно и во взаимодействии с другими явлениями, накапливая знания постепенно, время от времени возвращаясь к уже, казалось бы, понятому и с каждым новым витком все больше проникая в суть явления. Хорошим подходом является первоначальное изучение общих принципов некоторой области знаний с последующим детальным рассмотрением реализации этих принципов в конкретных методах, технологиях или конструкциях.

Первая часть книги является таким «первым витком» изучения компьютерных сетей. В этой части, состоящей из семи глав, описаны основные принципы и архитектурные решения, которые лежат в основе всех современных сетевых технологий, рассматриваемых в последующих частях книги. Следуя процессу конвергенции сетей, мы рассматривали принципы коммутации, мультиплексирования, маршрутизации, адресации и архитектуры сетей с наиболее общих позиций, сравнивая принципы организации компьютерных сетей с аналогичными принципами других телекоммуникационных сетей — телефонных, первичных, радио и телевизионных.

Завершает часть глава, посвященная проблемам качества обслуживания в пакетных сетях. Новая роль компьютерных сетей как основы для создания следующего поколения публичных сетей, предоставляющих все виды информационных услуг и переносящих данные, а также аудио- и видеотрафик, привела к проникновению методов обеспечения качества обслуживания практически во все коммуникационные технологии. Таким образом, концепции качества обслуживания, которые достаточно долго рассматривались как нетривиальное направление сетевой отрасли, вошли в число базовых принципов построения компьютерных сетей.

- Глава 1. Эволюция компьютерных сетей
- Глава 2. Общие принципы построения сетей
- Глава 3. Коммутация каналов и пакетов
- Глава 4. Архитектура и стандартизация сетей
- Глава 5. Примеры сетей
- Глава 6. Сетевые характеристики
- Глава 7. Методы обеспечения качества обслуживания

ГЛАВА 1 Эволюция компьютерных сетей

История любой отрасли науки и техники позволяет не только удовлетворить естественное любопытство, но и глубже понять сущность основных достижений в этой отрасли, осознать существующие тенденции и правильно оценить перспективность тех или иных направлений развития. Компьютерные сети появились сравнительно недавно, в конце 60-х годов прошлого столетия (правда, уточнение «прошлого столетия» прибавляет им вес и даже делает старше своих «тридцати с чем-то» лет). Естественно, что компьютерные сети унаследовали много полезных свойств от других, более старых и распространенных телекоммуникационных сетей, а именно телефонных. В то же время компьютерные сети привнесли в телекоммуникационный мир нечто совершенно новое — они сделали общедоступными неисчерпаемые объемы информации, созданные цивилизацией за несколько тысячелетий своего существования и продолжающие пополняться с растущей скоростью в наши дни.

Результатом влияния компьютерных сетей на остальные типы телекоммуникационных сетей стал процесс их конвергенции. Этот процесс начался достаточно давно, одним из первых признаков сближения была передача телефонными сетями голоса в цифровой форме. Компьютерные сети также активно идут навстречу телекоммуникационным сетям, разрабатывая новые сервисы, которые ранее были прерогативой телефонных, радио и телевизионных сетей — сервисы IP-телефонии, радио- и видеовещания, ряд других. Процесс конвергенции продолжается, и о том, каким будет его конечный результат, с уверенностью пока говорить рано. Однако понимание истории развития сетей, описываемой в данной главе, делает более понятными основные проблемы, стоящие перед разработчиками компьютерных сетей.

При написании этой главы авторы столкнулись с дилеммой: невозможно рассказывать об истории отрасли, не называя конкретные технологии и концепции. Но в то же время невозможно давать пояснения этих технологий и концепций, так как читатель, перелистывающий первые страницы, еще не готов к восприятию объяснений. Авторы пошли по пути компромисса, отложив на будущее исчерпывающие пояснения многих терминов ради того, чтобы в самом начале изучения компьютерных сетей читатель имел возможность представить картину эволюции компьютерных сетей во всем ее красочном многообразии. И, конечно, было бы очень полезно вернуться к этой главе, после того как будет перевернута последняя страница книги, чтобы, вооружась новыми знаниями, сделать качественно новую попытку оценить прошлое и будущее компьютерных сетей.

Два корня компьютерных сетей

Вычислительная и телекоммуникационная технологии

Компьютерные сети, которым посвящена данная книга, отнюдь не являются единственным видом сетей, созданным человеческой цивилизацией. Даже водопроводы Древнего Рима можно рассматривать как один из наиболее древних примеров сетей, покрывающих большие территории и обслуживающих многочисленных клиентов. Другой, менее экзотический пример — электрические сети. В них легко можно найти аналоги компонентов любой территориальной компьютерной сети: источникам информационных ресурсов соответствуют электростанции, магистралям — высоковольтные линии электропередачи, сетям доступа — трансформаторные подстанции, клиентским терминалам — осветительные и бытовые электроприборы.

Компьютерные сети, называемые также **сетями передачи данных**, являются логическим результатом эволюции двух важнейших научно-технических отраслей современной цивилизации — компьютерных и телекоммуникационных технологий.

С одной стороны, сети представляют собой частный случай распределенных вычислительных систем, в которых группа компьютеров согласованно решает набор взаимосвязанных задач, обмениваясь данными в автоматическом режиме. С другой стороны, компьютерные сети могут рассматриваться как средство передачи информации на большие расстояния, для чего в них применяются методы кодирования и мультиплексирования данных, получившие развитие в различных телекоммуникационных системах (рис. 1.1).

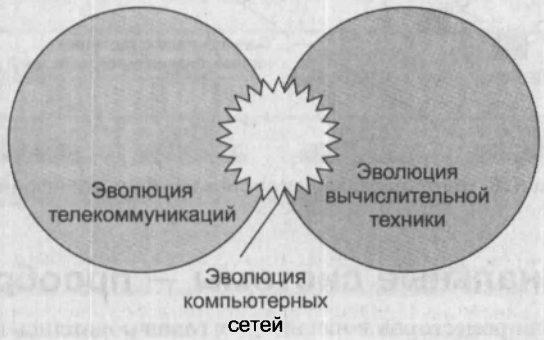


Рис. 1.1. Эволюция компьютерных сетей на стыке вычислительной техники и телекоммуникационных технологий

Системы пакетной обработки

Обратимся сначала к компьютерному корню вычислительных сетей. Первые компьютеры 50-х годов — большие, громоздкие и дорогие — предназначались для очень небольшого числа избранных пользователей. Часто эти монстры занимали целые здания. Такие ком-

пьютеры не были предназначены для интерактивной работы пользователя, а применялись в режиме пакетной обработки.

Системы пакетной обработки, как правило, строились на базе мэйнфрейма — мощного и надежного компьютера универсального назначения. Пользователи подготавливали перфокарты, содержащие данные и команды программ, и передавали их в вычислительный центр (рис. 1.2). Операторы вводили эти карты в компьютер, а распечатанные результаты пользователи получали обычно только на следующий день. Таким образом, одна неверно набитая карта означала как минимум суточную задержку. Конечно, для пользователей интерактивный режим работы, при котором можно с терминала оперативно руководить процессом обработки своих данных, был бы удобней. Но интересами пользователей на первых этапах развития вычислительных систем в значительной степени пренебрегали. Во главу угла ставилась эффективность работы самого дорогого устройства вычислительной машины — процессора, даже в ущерб эффективности работы использующих его специалистов.

Рис. 1.2. Централизованная система на базе мэйнфрейма

Многотерминальные системы — прообраз сети

По мере удешевления процессоров в начале 60-х годов появились новые способы организации вычислительного процесса, которые позволили учесть интересы пользователей. Начали развиваться интерактивные **многотерминальные системы разделения времени** (рис. 1.3). В таких системах каждый пользователь получал собственный терминал, с помощью которого он мог вести диалог с компьютером. Количество одновременно работающих с компьютером пользователей определялось его мощностью: время реакции вычислительной системы должно было быть достаточно мало, чтобы пользователю была не слишком заметна параллельная работа с компьютером других пользователей.

Терминалы, выйдя за пределы вычислительного центра, рассредоточились по всему предприятию. И хотя вычислительная мощность оставалась полностью централизованной,

некоторые функции, такие как ввод и вывод данных, стали распределенными. Подобные многотерминальные централизованные системы внешне уже были очень похожи на локальные вычислительные сети. Действительно, рядовой пользователь работу за терминалом мэйнфрейма воспринимал примерно так же, как сейчас он воспринимает работу за подключенным к сети персональным компьютером. Пользователь мог получить доступ к общим файлам и периферийным устройствам, при этом у него поддерживалась полная иллюзия единоличного владения компьютером, так как он мог запустить нужную ему программу в любой момент и почти сразу же получить результат. (Некоторые далекие от вычислительной техники пользователи даже были уверены, что все вычисления выполняются внутри их дисплея.)

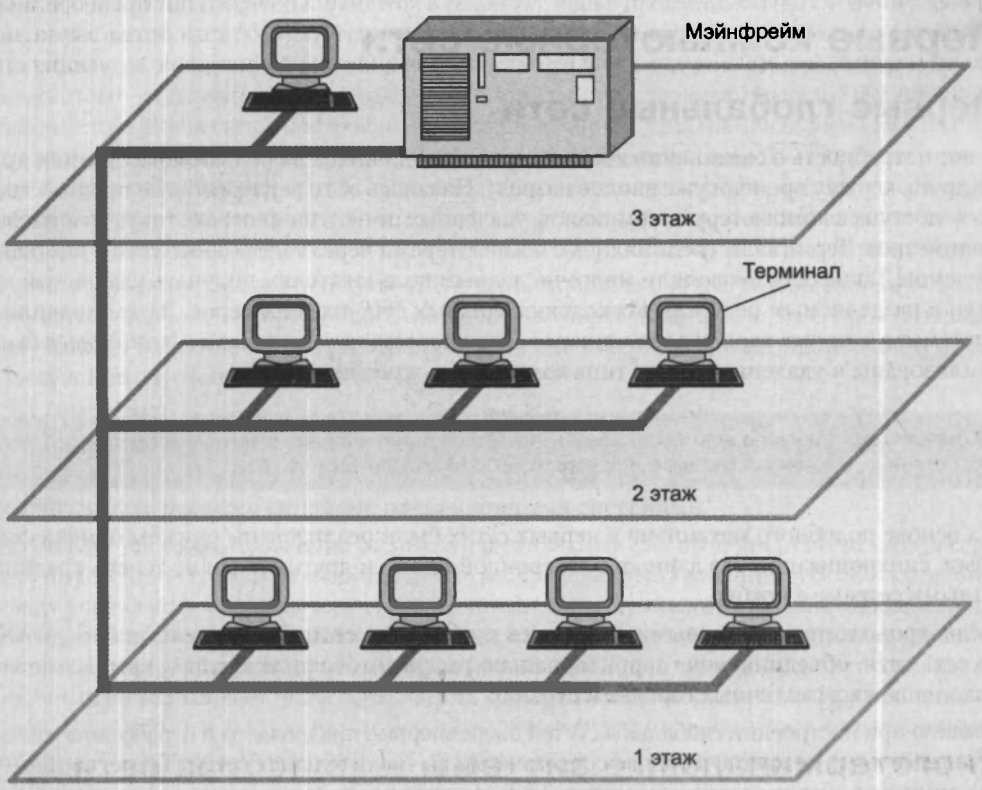


Рис. 1.3. Многотерминальная система — прообраз вычислительной сети

Многотерминальные системы, работающие в режиме разделения времени, стали первым шагом на пути создания локальных вычислительных сетей.

Однако до появления локальных сетей нужно было пройти еще большой путь, так как многотерминальные системы, хотя и имели внешние черты распределенных систем, все еще поддерживали централизованную обработку данных.

К тому же потребность предприятий в создании локальных сетей в это время еще не созрела — в одном здании просто нечего было объединять в сеть, так как из-за высокой стоимости вычислительной техники предприятия не могли себе позволить роскошь приобретения нескольких компьютеров. В этот период был справедлив так называемый **закон Гроша**, который эмпирически отражал уровень технологии того времени. В соответствии с этим законом производительность компьютера была пропорциональна квадрату его стоимости, отсюда следовало, что за одну и ту же сумму было выгоднее купить одну мощную машину, чем две менее мощных — их суммарная мощность оказывалась намного ниже мощности дорогой машины.

Первые компьютерные сети

Первые глобальные сети

А вот потребность в соединении компьютеров, находящихся на большом расстоянии друг от друга, к этому времени уже вполне назрела. Началось все с решения более простой задачи — доступа к компьютеру с терминалов, удаленных от него на многие сотни, а то и тысячи километров. Терминалы соединялись с компьютерами через телефонные сети с помощью модемов. Такие сети позволяли многочисленным пользователям получать удаленный доступ к разделяемым ресурсам нескольких мощных суперкомпьютеров. Затем появились системы, в которых наряду с удаленными соединениями типа *терминал–компьютер* были реализованы и удаленные связи типа *компьютер–компьютер*.

Компьютеры получили возможность обмениваться данными в автоматическом режиме, что, собственно, и является базовым признаком любой вычислительной сети.

На основе подобного механизма в первых сетях были реализованы службы обмена файлами, синхронизации баз данных, электронной почты и другие ставшие теперь традиционными сетевые службы.

Итак, хронологически первыми появились **глобальные сети** (Wide Area Network, WAN), то есть сети, объединяющие территориально рассредоточенные компьютеры, возможно находящиеся в различных городах и странах.

Именно при построении глобальных сетей были впервые предложены и отработаны многие основные идеи, лежащие в основе современных вычислительных сетей. Такие, например, как многоуровневое построение коммуникационных протоколов, концепции коммутации и маршрутизации пакетов.

Глобальные компьютерные сети очень многое унаследовали от других, гораздо более старых и распространенных глобальных сетей — *телефонных*. Главное технологическое новшество, которое привнесли с собой первые глобальные компьютерные сети, состояло в отказе от принципа коммутации каналов, на протяжении многих десятков лет успешно использовавшегося в телефонных сетях.

Выделяемый на все время сеанса связи составной телефонный канал, передающий информацию с постоянной скоростью, не мог эффективно использоваться пульсирующим трафиком компьютерных данных, у которого периоды интенсивного обмена чередуются с продолжительными паузами. Натурные эксперименты и математическое моделирование показали, что

пульсирующий и в значительной степени не чувствительный к задержкам компьютерный трафик гораздо эффективней передается сетями, работающими по принципу коммутации пакетов, когда данные разделяются на небольшие порции — пакеты, — которые самостоятельно перемещаются по сети благодаря наличию адреса конечного узла в заголовке пакета.

Так как прокладка высококачественных линий связи на большие расстояния обходится очень дорого, то в первых глобальных сетях часто использовались уже существующие каналы связи, изначально предназначенные совсем для других целей. Например, в течение многих лет глобальные сети строились на основе телефонных каналов тональной частоты, способных в каждый момент времени вести передачу только одного разговора в аналоговой форме. Поскольку скорость передачи дискретных компьютерных данных по таким каналам была очень низкой (десятки килобитов в секунду), набор предоставляемых услуг в глобальных сетях такого типа обычно ограничивался передачей файлов (преимущественно в фоновом режиме) и электронной почтой. Помимо низкой скорости такие каналы имеют и другой недостаток — они вносят значительные искажения в передаваемые сигналы. Поэтому протоколы глобальных сетей, построенных с использованием каналов связи низкого качества, отличаются сложными процедурами контроля и восстановления данных. Типичным примером таких сетей являются сети X.25, разработанные еще в начале 70-х, когда низкоскоростные аналоговые каналы, арендуемые у телефонных компаний, были преобладающим типом каналов, соединяющих компьютеры и коммутаторы глобальной вычислительной сети.

В 1969 году министерство обороны США инициировало работы по объединению в единую сеть суперкомпьютеров оборонных и научно-исследовательских центров. Эта сеть, получившая название ARPANET, стала отправной точкой для создания первой и самой известной ныне глобальной сети — Интернет.

Сеть ARPANET объединяла компьютеры разных типов, работавшие под управлением различных операционных систем (ОС) с дополнительными модулями, реализующими коммуникационные протоколы, общие для всех компьютеров сети. ОС этих компьютеров можно считать *первыми* **сетевыми операционными системами**.

Истинно сетевые ОС в отличие от многотерминальных ОС позволяли не только рассредоточить пользователей, но и организовать распределенные хранение и обработку данных между несколькими компьютерами, связанными электрическими связями. Любая сетевая операционная система, с одной стороны, выполняет все функции локальной операционной системы, а с другой стороны, обладает некоторыми дополнительными средствами, позволяющими ей взаимодействовать через сеть с операционными системами других компьютеров. Программные модули, реализующие сетевые функции, появлялись в операционных системах постепенно, по мере развития сетевых технологий, аппаратной базы компьютеров и возникновения новых задач, требующих сетевой обработки.

Прогресс глобальных компьютерных сетей во многом определялся прогрессом телефонных сетей.

С конца 60-х годов в телефонных сетях все чаще стала применяться передача голоса в цифровой форме.

Это привело к появлению высокоскоростных цифровых каналов, соединяющих автоматические телефонные станции (АТС) и позволяющих одновременно передавать десятки и сотни разговоров.

К настоящему времени глобальные сети по разнообразию и качеству предоставляемых услуг догнали локальные сети, которые долгое время лидировали в этом отношении, хотя и появились на свет значительно позже.

Первые локальные сети

Важное событие, повлиявшее на эволюцию компьютерных сетей, произошло в начале 70-х годов. В результате технологического прорыва в области производства компьютерных компонентов появились **большие интегральные схемы** (БИС). Их сравнительно невысокая стоимость и хорошие функциональные возможности привели к созданию мини-компьютеров, которые стали реальными конкурентами мэйнфреймов. Эмпирический закон Гроша перестал соответствовать действительности, так как десяток мини-компьютеров, имея ту же стоимость, что и мэйнфрейм, решали некоторые задачи (как правило, хорошо распараллеливаемые) быстрее.

Даже небольшие подразделения предприятий получили возможность иметь собственные компьютеры. Мини-компьютеры решали задачи управления технологическим оборудованием, складом и другие задачи уровня отдела предприятия. Таким образом, появилась концепция распределения компьютерных ресурсов по всему предприятию. Однако при этом все компьютеры одной организации по-прежнему продолжали работать *автономно* (рис. 1.4).

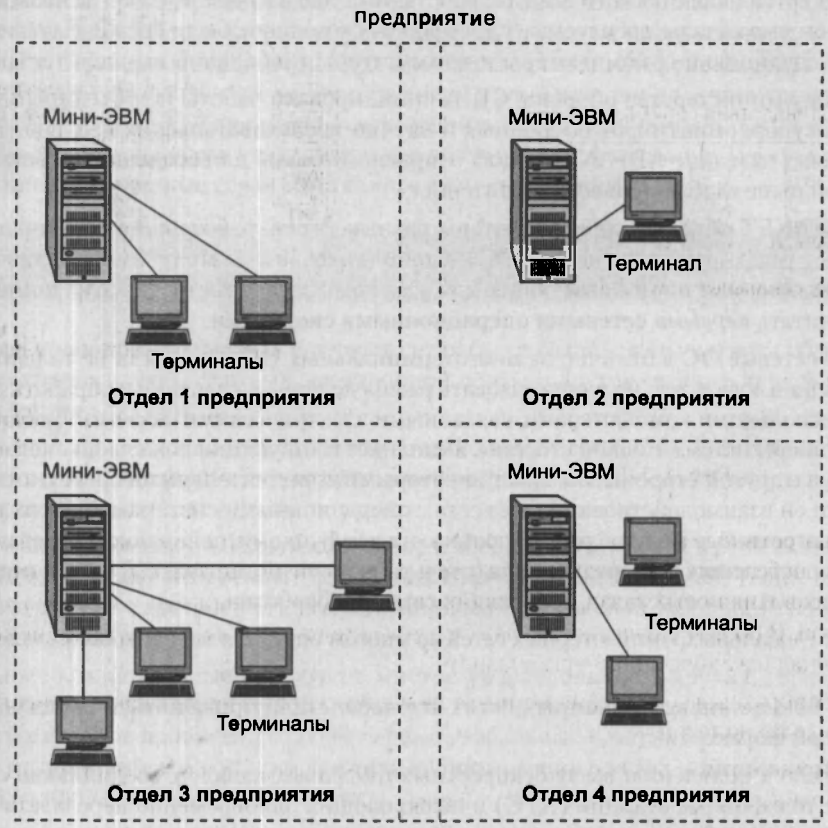


Рис. 1.4. Автономное использование нескольких мини-компьютеров на одном предприятии

Шло время, и потребности пользователей вычислительной техники росли. Их уже не удовлетворяла изолированная работа на собственном компьютере, им хотелось в автоматическом режиме обмениваться компьютерными данными с пользователями других подразделений. Ответом на эту потребность стало появление первых локальных вычислительных сетей (рис. 1.5).

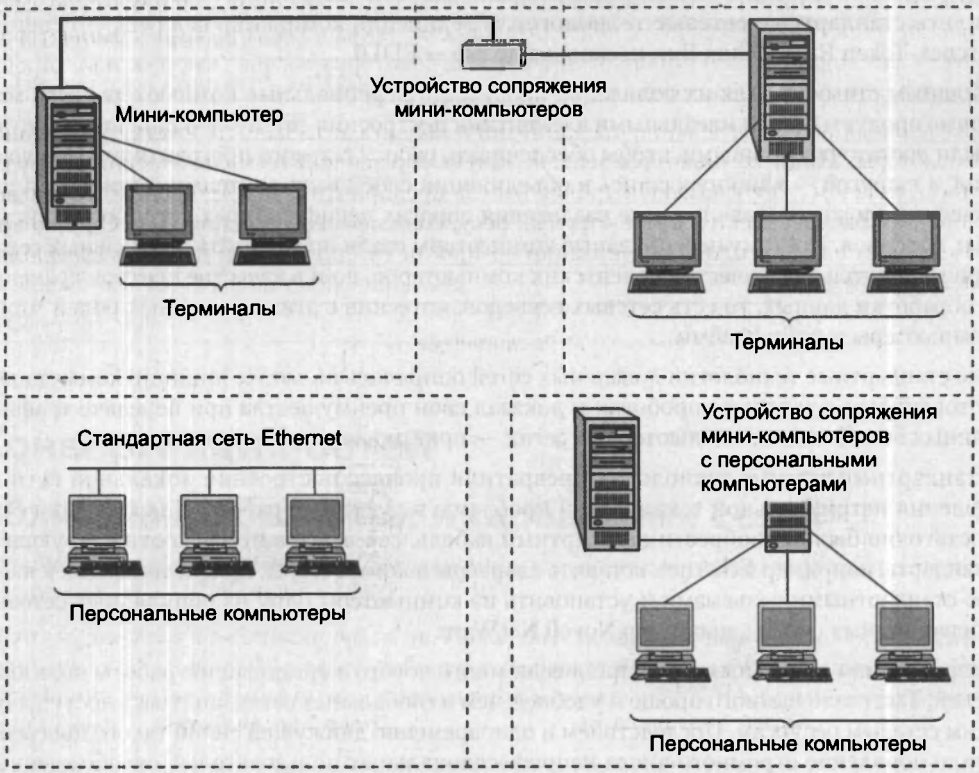


Рис. 1.5. Различные типы связей в первых локальных сетях

Локальные сети (Local Area Network, LAN) — это объединения компьютеров, сосредоточенных на небольшой территории, обычно в радиусе не более 1–2 км, хотя в отдельных случаях локальная сеть может иметь и большие размеры, например несколько десятков километров. В общем случае локальная сеть представляет собой коммуникационную систему, принадлежащую одной организации.

На первых порах для соединения компьютеров друг с другом использовались нестандартные сетевые технологии.

Сетевая технология — это согласованный набор программных и аппаратных средств (например, драйверов, сетевых адаптеров, кабелей и разъемов), а также механизмов передачи данных по линиям связи, достаточный для построения вычислительной сети.

Разнообразные устройства сопряжения, использующие собственные способы представления данных на линиях связи, свои типы кабелей и т. п., могли соединять только те

конкретные модели компьютеров, для которых были разработаны, например, мини-компьютеры PDP-11 с мэйнфреймом IBM 360 или мини-компьютеры HP с микрокомпьютерами LSI-11. Такая ситуация создала большой простор для творчества студентов — названия многих курсовых и дипломных проектов начинались тогда со слов «Устройство сопряжения...».

В середине 80-х годов положение дел в локальных сетях кардинально изменилось. Утвердились **стандартные сетевые технологии** объединения компьютеров в сеть — Ethernet, Arcnet, Token Ring, Token Bus, несколько позже — FDDI.

Мощным стимулом для их появления послужили **персональные компьютеры**. Эти массовые продукты стали идеальными элементами построения сетей — с одной стороны, они были достаточно мощными, чтобы обеспечивать работу сетевого программного обеспечения, а с другой — явно нуждались в объединении своей вычислительной мощности для решения сложных задач, а также разделения дорогих периферийных устройств и дисковых массивов. Поэтому персональные компьютеры стали преобладать в локальных сетях, причем не только в качестве клиентских компьютеров, но и в качестве центров хранения и обработки данных, то есть сетевых серверов, потеснив с этих привычных ролей мини-компьютеры и мэйнфреймы.

Все стандартные технологии локальных сетей опирались на тот же принцип коммутации, который был с успехом опробован и доказал свои преимущества при передаче трафика данных в глобальных компьютерных сетях, — принцип коммутации пакетов.

Стандартные сетевые технологии превратили процесс построения локальной сети из решения нетривиальной технической проблемы в рутинную работу. Для создания сети достаточно было приобрести стандартный кабель, сетевые адаптеры соответствующего стандарта, например Ethernet, вставить адаптеры в компьютеры, присоединить их к кабелю стандартными разъемами и установить на компьютеры одну из популярных сетевых операционных систем, например Novell NetWare.

Разработчики локальных сетей привнесли много нового в организацию работы пользователей. Так, стало намного проще и удобнее, чем в глобальных сетях, получать доступ к общим сетевым ресурсам. Последствием и одновременно движущей силой такого прогресса стало появление огромного числа непрофессиональных пользователей, освобожденных от необходимости изучать специальные (и достаточно сложные) команды для сетевой работы.

Конец 90-х выявил явного лидера среди технологий локальных сетей — семейство Ethernet, в которое вошли классическая технология Ethernet со скоростью передачи 10 Мбит/с, а также Fast Ethernet со скоростью 100 Мбит/с и Gigabit Ethernet со скоростью 1000 Мбит/с.

Простые алгоритмы работы предопределяют низкую стоимость оборудования Ethernet. Широкий диапазон иерархии скоростей позволяет рационально строить локальную сеть, выбирая ту технологию семейства, которая в наибольшей степени отвечает задачам предприятия и потребностям пользователей. Важно также, что все технологии Ethernet очень близки друг к другу по принципам работы, что упрощает обслуживание и интеграцию этих сетей.

Хронологическую последовательность важнейших событий, ставших историческими вехами на пути появления первых компьютерных сетей, иллюстрирует табл. 1.1.

Таблица 1.1. Хронология важнейших событий на пути появления первых компьютерных сетей

Этап	Время
Первые глобальные связи компьютеров, первые эксперименты с пакетными сетями	Конец 60-х
Начало передач по телефонным сетям голоса в цифровой форме	Конец 60-х
Появление больших интегральных схем, первые мини-компьютеры, первые нестандартные локальные сети	Начало 70-х
Создание сетевой архитектуры IBM SNA	1974
Стандартизация технологии X.25	1974
Появление персональных компьютеров, создание Интернета в современном виде, установка на всех узлах стека TCP/IP	Начало 80-х
Появление стандартных технологий локальных сетей (Ethernet — 1980 г., Token Ring, FDDI — 1985 г.)	Середина 80-х
Начало коммерческого использования Интернета	Конец 80-х
Изобретение Web	1991

Конвергенция сетей

Сближение локальных и глобальных сетей

В конце 80-х годов отличия между локальными и глобальными сетями проявлялись весьма отчетливо.

Протяженность и качество линий связи. Локальные компьютерные сети по определению отличаются от глобальных сетей небольшими расстояниями между узлами сети. Это в принципе делает возможным использование в локальных сетях более качественных линий связи.

- *Сложность методов передачи данных.* В условиях низкой надежности физических каналов в глобальных сетях требуются более сложные, чем в локальных сетях, методы передачи данных и соответствующее оборудование.
- *Скорость обмена данными* в локальных сетях (10, 16 и 100 Мбит/с) в то время была существенно выше, чем в глобальных (от 2,4 Кбит/с до 2 Мбит/с).
- *Разнообразие услуг.* Высокие скорости обмена данными позволили предоставлять в локальных сетях широкий спектр услуг — это, прежде всего, разнообразные механизмы использования файлов, хранящихся на дисках других компьютеров сети, совместное использование устройств печати, модемов, факсов, доступ к единой базе данных, электронная почта и др. В то же время глобальные сети в основном ограничивались почтовыми и файловыми услугами в их простейшем (не самом удобном для пользователя) виде.

Постепенно различия между локальными и глобальными сетевыми технологиями стали сглаживаться. Изолированные ранее локальные сети начали объединять друг с другом, при этом в качестве связующей среды использовались глобальные сети. Тесная интеграция

локальных и глобальных сетей привела к значительному взаимопроникновению соответствующих технологий.

Сближение в методах передачи данных происходит на платформе цифровой передачи данных по волоконно-оптическим линиям связи. Эта среда передачи используется практически во всех технологиях локальных сетей для скоростного обмена информацией на расстояниях свыше 100 метров, на ней же построены современные магистрали первичных сетей SDH и DWDM, предоставляющих свои цифровые каналы для объединения оборудования глобальных компьютерных сетей.

Высокое качество цифровых каналов изменило требования к протоколам глобальных компьютерных сетей. На первый план вместо процедур обеспечения надежности вышли процедуры обеспечения гарантированной средней скорости доставки информации пользователям, а также механизмы приоритетной обработки пакетов особенно чувствительного к задержкам трафика, например голосового. Эти изменения нашли отражение в новых технологиях глобальных сетей, таких как Frame Relay и ATM. В этих сетях предполагается, что искажение битов происходит настолько редко, что ошибочный пакет выгоднее просто уничтожить, а все проблемы, связанные с его потерей, перепоручить программному обеспечению более высокого уровня, которое непосредственно не входит в состав сетей Frame Relay и ATM.

Большой вклад в сближение локальных и глобальных сетей внесло доминирование протокола IP. Этот протокол сегодня работает поверх любых технологий локальных и глобальных сетей (Ethernet, Token Ring, ATM, Frame Relay), объединяя различные подсети в единую составную сеть.

Начиная с 90-х годов компьютерные глобальные сети, работающие на основе скоростных цифровых каналов, существенно расширили спектр предоставляемых услуг и догнали в этом отношении локальные сети. Стало возможным создание служб, работа которых связана с доставкой пользователю больших объемов информации в реальном времени — изображений, видеофильмов, голоса, в общем, всего того, что получило название мультимедийной информации. Наиболее яркий пример — гипертекстовая информационная служба World Wide Web, ставшая основным поставщиком информации в Интернете. Ее интерактивные возможности превзошли возможности многих аналогичных служб локальных сетей, так что разработчикам локальных сетей пришлось просто позаимствовать эту службу у глобальных сетей. Процесс переноса технологий из глобальной сети Интернет в локальные приобрел такой массовый характер, что появился даже специальный термин — **intranet-технологии** (intra — внутренний).

В локальных сетях в последнее время уделяется такое же большое внимание методам обеспечения защиты информации от несанкционированного доступа, как и в глобальных. Это обусловлено тем, что локальные сети перестали быть изолированными, чаще всего они имеют выход в «большой мир» через глобальные связи.

И наконец, появляются новые технологии, изначально предназначенные для обоих видов сетей. Ярким представителем нового поколения технологий является технология ATM, которая может служить основой как глобальных, так и локальных сетей, эффективно объединяя все существующие типы трафика в одной транспортной сети. Другим примером является семейство технологий Ethernet, имеющее явные «локальные» корни. Новый стандарт Ethernet 10G, позволяющий передавать данные со скоростью 10 Гбит/с, предназначен для магистралей как глобальных, так и крупных локальных сетей.

Еще одним признаком сближения локальных и глобальных сетей является появление сетей, занимающих промежуточное положение между локальными и глобальными сетями.

Городские сети, или **сети мегаполисов** (Metropolitan Area Network, MAN), предназначены для обслуживания территории крупного города.

Эти сети используют цифровые линии связи, часто оптоволоконные, со скоростями на магистрали от 155 Мбит/с и выше. Они обеспечивают экономичное соединение локальных сетей между собой, а также выход в глобальные сети. Сети MAN первоначально были разработаны только для передачи данных, но сейчас перечень предоставляемых ими услуг расширился, в частности они поддерживают видеоконференции и интегральную передачу голоса и текста. Современные сети MAN отличаются разнообразием предоставляемых услуг, позволяя своим клиентам объединять коммуникационное оборудование различного типа, в том числе офисные АТС.

Конвергенция компьютерных и телекоммуникационных сетей

С каждым годом усиливается тенденция сближения компьютерных и телекоммуникационных сетей разных видов. Предпринимаются попытки создания универсальной, так называемой **мультисервисной сети**, способной предоставлять услуги как компьютерных, так и телекоммуникационных сетей.

К телекоммуникационным сетям относятся телефонные сети, радиосети и телевизионные сети. Главное, что объединяет их с компьютерными сетями, — то, что в качестве ресурса, предоставляемого клиентам, выступает информация. Однако имеется некоторая специфика, касающаяся вида, в котором представляют информацию компьютерные и телекоммуникационные сети. Так, изначально компьютерные сети разрабатывались для передачи алфавитно-цифровой информации, которую часто называют просто *данными*, поэтому у компьютерных сетей имеется и другое название — **сети передачи данных**, в то время как телекоммуникационные сети были созданы для передачи только *голосовой информации* (и изображения в случае телевизионных сетей).

Сегодня мы являемся свидетелями конвергенции телекоммуникационных и компьютерных сетей, которая идет по нескольким направлениям.

Прежде всего, наблюдается *сближение видов услуг*, предоставляемых клиентам. Первая и не очень успешная попытка создания мультисервисной сети, способной оказывать различные услуги, в том числе услуги телефонии и передачи данных, привела к появлению технологии **цифровых сетей с интегрированным обслуживанием** (Integrated Services Digital Network, ISDN). Однако на практике ISDN предоставляет сегодня в основном телефонные услуги, а на роль глобальной **мультисервисной сети нового поколения**, часто называемой в англоязычной литературе Next Generation Network (NGN), или New Public Network (NPN), претендует Интернет. Интернет будущего должен обладать возможностью оказывать все виды телекоммуникационных услуг, в том числе новые виды комбинированных услуг, в которых сочетаются несколько традиционных услуг, например услуга универсальной службы сообщений, объединяющей электронную почту, телефонию, факсимильную службу и пейджинговую связь. Наибольших успехов на практическом поприще достигла IP-телефония, услугами которой прямо или косвенно сегодня пользуются миллионы людей. Однако для того чтобы стать сетью NGN, Интернету еще предстоит пройти большой путь.

Технологическое сближение сетей происходит сегодня на основе цифровой передачи информации различного типа, метода коммутации пакетов и программирования услуг.

Телефония уже давно сделала ряд шагов навстречу компьютерным сетям, прежде всего, за счет представления голоса в цифровой форме, что делает принципиально возможным передачу телефонного и компьютерного трафика по одним и тем же цифровым каналам (телевидение также может сегодня передавать изображение в цифровой форме). Телефонные сети широко используют комбинацию методов коммутации каналов и пакетов. Так, для передачи служебных сообщений (называемых сообщениями сигнализации) применяются протоколы коммутации пакетов, аналогичные протоколам компьютерных сетей, а для передачи собственно голоса между абонентами коммутируется традиционный составной канал.

Дополнительные услуги телефонных сетей, такие как переадресация вызова, конференц-связь, телеголосование и другие, могут создаваться с помощью так называемой **интеллектуальной сети** (Intelligent Network, IN), по своей сути являющейся компьютерной сетью с серверами, на которых программируется логика услуг.

Сегодня пакетные методы коммутации постепенно теснят традиционные для телефонных сетей методы коммутации каналов даже при передаче голоса. У этой тенденции есть достаточно очевидная причина — на основе метода коммутации пакетов можно более эффективно использовать пропускную способность каналов связи и коммутационного оборудования. Например, паузы в телефонном разговоре могут составлять до 40 % общего времени соединения, однако только пакетная коммутация позволяет «вырезать» паузы и использовать высвободившуюся пропускную способность канала для передачи трафика других абонентов. Другой веской причиной перехода к коммутации пакетов является популярность Интернета — сети, построенной на основе данной технологии.

Обращение к технологии коммутации пакетов для одновременной передачи через пакетные сети разнородного трафика — голоса, видео и текста — сделало актуальным разработку новых методов обеспечения требуемого **качества обслуживания** (Quality of Service, QoS). Методы QoS призваны минимизировать уровень задержек для чувствительного к ним трафика, например голосового, и одновременно гарантировать среднюю скорость и динамичную передачу пульсаций для трафика данных.

Однако неверно было бы говорить, что методы коммутации каналов морально устарели и у них нет будущего. На новом витке спирали развития они находят свое применение, но уже в новых технологиях.

Компьютерные сети тоже многое позаимствовали у телефонных и телевизионных сетей. В частности, они берут на вооружение методы обеспечения отказоустойчивости телефонных сетей, за счет которых последние демонстрируют высокую степень надежности, так недостающую порой Интернету и корпоративным сетям.

Сегодня становится все более очевидным, что мультисервисная сеть нового поколения не может быть создана в результате «победы» какой-нибудь одной технологии или одного подхода. Ее может породить только процесс конвергенции, когда от каждой технологии будет взято все самое лучшее и соединено в некоторый новый сплав, который и даст требуемое качество для поддержки существующих и создания новых услуг. Появился новый термин — **инфокоммуникационная сеть**, который прямо говорит о двух составляющих современной сети — информационной (компьютерной) и телекоммуникационной. Учитывая, что новый термин еще не приобрел достаточной популярности, мы будем использовать устоявшийся термин «телекоммуникационная сеть» в расширенном значении, то есть включать в него и компьютерные сети.

Выводы

Компьютерные сети стали логическим результатом эволюции компьютерных и телекоммуникационных технологий. С одной стороны, они являются частным случаем распределенных компьютерных систем, а с другой — могут рассматриваться как средство передачи информации на большие расстояния, для чего в них применяются методы кодирования и мультиплексирования данных, получившие развитие в различных телекоммуникационных системах.

Классифицируя сети по территориальному признаку, различают глобальные (WAN), локальные (LAN) и городские (MAN) сети.

Хронологически первыми появились сети WAN. Они объединяют компьютеры, рассредоточенные на расстоянии сотен и тысяч километров. Первые глобальные компьютерные сети очень многое унаследовали от телефонных сетей. В них часто использовались уже существующие и не очень качественные линии связи, что приводило к низким скоростям передачи данных и ограничивало набор предоставляемых услуг передачей файлов в фоновом режиме и электронной почтой.

Сети LAN ограничены расстояниями в несколько километров; они строятся с использованием высококачественных линий связи, которые позволяют, применяя более простые методы передачи данных, чем в глобальных сетях, достигать высоких скоростей обмена данными — до нескольких гигабитов в секунду. Услуги предоставляются в режиме подключения и отличаются разнообразием.

Сети MAN предназначены для обслуживания территории крупного города. При достаточно больших расстояниях между узлами (десятки километров) они обладают качественными линиями связи и поддерживают высокие скорости обмена. Сети MAN обеспечивают экономичное соединение локальных сетей между собой, а также доступ к глобальным сетям.

Важнейший этап в развитии сетей — появление стандартных сетевых технологий: Ethernet, FDDI, Token Ring, позволяющих быстро и эффективно объединять компьютеры различных типов.

В конце 80-х годов локальные и глобальные сети имели существенные отличия по протяженности и качеству линий связи, сложности методов передачи данных, скорости обмена данными, разнообразию предоставляемых услуг и масштабируемости. В дальнейшем в результате тесной интеграции LAN, WAN и MAN произошло взаимопроникновение соответствующих технологий.

Вопросы и задания

1. Что было унаследовано компьютерными сетями от вычислительной техники, а что от телефонных сетей?
2. Какие свойства многотерминальной системы отличают ее от компьютерной сети?
3. Когда впервые были получены значимые практические результаты по объединению компьютеров с помощью глобальных связей?
4. Что такое ARPANET?
5. Какое из следующих событий произошло позже других:

   а) изобретение Web;

   б) появление стандартных технологий LAN;

   в) начало передачи голоса в цифровой форме по телефонным сетям.
6. Какое событие послужило стимулом к активизации работ по созданию LAN?
7. Когда была стандартизована технология Ethernet?

8. По каким направлениям идет сближение компьютерных и телекоммуникационных сетей.
9. Поясните термины «мультисервисная сеть», «инфокоммуникационная сеть», «интеллектуальная сеть».
10. Поясните, почему сети WAN появились раньше, чем сети LAN.
11. Найдите исторические связи между технологией X.25 и сетью ARPANET, пользуясь источниками информации в Интернете.
12. Считаете ли вы, что история компьютерных сетей может быть сведена к истории Интернета? Обоснуйте свое мнение.

ГЛАВА 2 Общие принципы построения сетей

Когда вы приступаете к изучению конкретных технологий для сетей LAN, WAN и MAN, таких как Ethernet, IP или ATM, то очень скоро начинаете понимать, что у этих технологий есть много общего. При этом они не являются тождественными, в каждой технологии и протоколе есть свои особенности, так что нельзя механически перенести знания из одной технологии в другую.

Изучение общих принципов построения компьютерных сетей поможет вам в дальнейшем быстрее «разбираться» с любой конкретной сетевой технологией. Однако известное высказывание «Знание нескольких принципов освобождает от запоминания множества фактов» не стоит воспринимать буквально — хороший специалист, конечно же, должен знать множество деталей и фактов. Знание принципов позволяет систематизировать эти частные сведения, связать их друг с другом в стройную систему и тем самым использовать более осознано и эффективно. Конечно, изучение принципов перед изучением конкретных технологий — задача непростая, особенно для читателей с практическим складом ума. Кроме того, всегда есть опасность неверного понимания какого-нибудь общего утверждения без проверки его в практической реализации. Поэтому мы просим читателей поверить нам пока на слово, что игра стоит свеч, а также последовать нашему совету: в ходе изучения материала последующих глав книги время от времени мысленно возвращайтесь к теоретическим вопросам и проверяйте себя, так ли вы понимали те или иные механизмы, когда изучали их впервые.

В этой главе мы рассмотрим такие фундаментальные понятия сетевых технологий, как коммутация и маршрутизация, мультиплексирование и разделение передающей среды. Мы познакомимся также с общими подходами, применяющимися при адресации узлов сети и выборе топологии.

Простейшая сеть из двух компьютеров

Совместное использование ресурсов

Исторически главной целью объединения компьютеров в сеть было *разделение ресурсов*: пользователи компьютеров, подключенных к сети, или приложения, выполняемые на этих компьютерах, получают возможность автоматического доступа к разнообразным ресурсам остальных компьютеров сети, к числу которых относятся:

- периферийные устройства, такие как диски, принтеры, плоттеры, сканеры и др.;
- данные, хранящиеся в оперативной памяти или на внешних запоминающих устройствах;
- вычислительная мощность (за счет удаленного запуска «своих» программ на «чужих» компьютерах).

Чтобы обеспечить пользователей разных компьютеров возможностью совместного использования ресурсов сети, компьютеры необходимо оснастить некими дополнительными *сетевыми* средствами.

Рассмотрим простейшую сеть, состоящую из двух компьютеров, к одному из которых подключен принтер (рис. 2.1). Какие дополнительные средства должны быть предусмотрены в обоих компьютерах, чтобы с принтером мог работать не только пользователь компьютера *B*, к которому этот принтер непосредственно подключен, но и пользователь компьютера *A*?

Рис. 2.1. Простейшая сеть

Сетевые интерфейсы

Для связи устройств в них, прежде всего, должны быть предусмотрены внешние[1] интерфейсы.

Интерфейс — в широком смысле — формально определенная логическая и/или физическая граница между взаимодействующими независимыми объектами. Интерфейс задает параметры, процедуры и характеристики взаимодействия объектов.

[1] Наряду с внешними электронные устройства могут использовать внутренние интерфейсы, определяющие логические и физические границы между входящими в их состав модулями. Так, известный интерфейс «общая шина» является внутренним интерфейсом компьютера, связывающим оперативную память, процессор и другие блоки компьютера.

Разделяют физический и логический интерфейсы.

- **Физический интерфейс** (называемый также **портом**) — определяется набором электрических связей и характеристиками сигналов. Обычно он представляет собой разъем с набором контактов, каждый из которых имеет определенное назначение, например, это может быть группа контактов для передачи данных, контакт синхронизации данных и т. п. Пара разъемов соединяется **кабелем**, состоящим из набора проводов, каждый из которых соединяет соответствующие контакты. В таких случаях говорят о создании **линии**, или **канала**, **связи** между двумя устройствами.
- **Логический интерфейс** (называемый также **протоколом**) — это набор информационных сообщений определенного формата, которыми обмениваются два устройства или две программы, а также набор правил, определяющих логику обмена этими сообщениями.

На рис. 2.2 мы видим интерфейсы двух типов: компьютер—компьютер и компьютер—периферийное устройство.

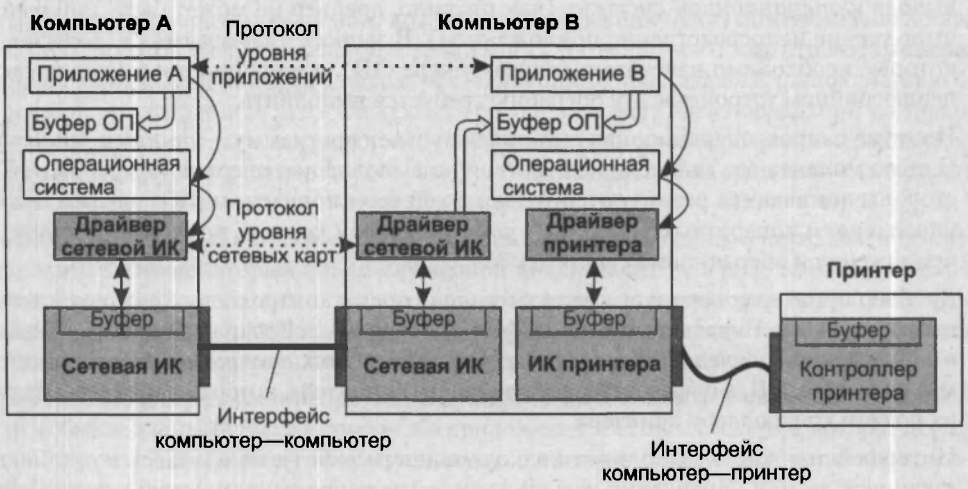


Рис. 2.2. Совместное использование принтера в компьютерной сети

- *Интерфейс компьютер—компьютер* позволяет двум компьютерам обмениваться информацией. С каждой стороны он реализуется парой:
  - аппаратным модулем, называемым **сетевым адаптером**, или **сетевой интерфейсной картой** (Network Interface Card, NIC);
  - **драйвером сетевой интерфейсной карты** — специальной программой, управляющей работой сетевой интерфейсной карты.
- *Интерфейс компьютер—периферийное устройство* (в данном случае интерфейс компьютер—принтер) позволяет компьютеру управлять работой периферийного устройства (ПУ). Этот интерфейс реализуется:
  - со стороны компьютера — **интерфейсной картой** и **драйвером ПУ** (принтера), подобным сетевой интерфейсной карте и ее драйверу;

- со стороны ПУ — **контроллером ПУ** (принтера), обычно представляющий собой аппаратное устройство[1], принимающее от компьютера как *данные*, например байты информации, которую нужно распечатать на бумаге, так и *команды*, которые он отрабатывает, управляя электромеханическими частями периферийного устройства, например выталкивая лист бумаги из принтера или перемещая магнитную головку диска.

Связь компьютера с периферийным устройством

Для того чтобы решить задачу организации доступа приложения, выполняемого на компьютере *A*, к ПУ через сеть, давайте, прежде всего, посмотрим, как управляет этим устройством приложение, выполняемое на компьютере *B*, к которому данное ПУ подключено непосредственно (см. рис. 2.2).

1. Пусть приложению *B* в какой-то момент потребовалось вывести на печать некоторые данные. Для этого приложение обращается с запросом на выполнение операции ввода-вывода к *операционной системе* (как правило, драйвер не может быть запущен на выполнение непосредственно приложением). В запросе указываются адрес данных, которые необходимо напечатать (адрес буфера ОП), и информация о том, на каком периферийном устройстве эту операцию требуется выполнить.
2. Получив запрос, операционная система запускает программу — *драйвер принтера*. С этого момента все дальнейшие действия по выполнению операции ввода-вывода со стороны компьютера реализуются только драйвером принтера и работающим под его управлением аппаратным модулем — *интерфейсной картой принтера* без участия приложения и операционной системы.
3. Драйвер принтера оперирует командами, понятными контроллеру принтера, такими, например, как «Печать символа», «Перевод строки», «Возврат каретки». Драйвер в определенной последовательности загружает коды этих команд, а также данные, взятые из буфера ОП, в буфер интерфейсной карты принтера, которая побайтно передает их по сети контроллеру принтера.
4. Интерфейсная карта выполняет низкоуровневую работу, не вдаваясь в детали, касающиеся логики управления устройством, смысла данных и команд, передаваемых ей драйвером, считая их однородным потоком байтов. После получения от драйвера очередного байта интерфейсная карта просто последовательно передает биты в линию связи, представляя каждый бит электрическим сигналом. Чтобы контроллеру принтера стало понятно, что начинается передача байта, перед передачей первого бита информационная карта формирует **стартовый сигнал** специфической формы, а после передачи последнего информационного бита — **стоповый сигнал**. Эти сигналы синхронизируют передачу байта. Контроллер, опознав стартовый бит, начинает принимать информационные биты, формируя из них байт в своем приемном буфере. Помимо информационных битов карта может передавать бит контроля четности для повышения достоверности обмена. При корректно выполненной передаче в буфере принтера устанавливается соответствующий признак.

---

[1] Встречаются и программно-управляемые контроллеры, например, для управления современными принтерами, обладающими сложной логикой.

5. Получив очередной байт, контроллер интерпретирует его и запускает заданную операцию принтера. Закончив работу по печати всех символов документа, драйвер принтера сообщает операционной системе о выполнении запроса, а та, в свою очередь, сигнализирует об этом событии приложению.

Обмен данными между двумя компьютерами

Механизмы взаимодействия компьютеров в сети многое позаимствовали у схемы взаимодействия компьютера с периферийными устройствами. В самом простом случае связь компьютеров может быть реализована с помощью тех же самых средств, которые используются для связи компьютера с периферией, с той разницей, что в этом случае активную роль играют обе взаимодействующие стороны.

Приложения *A* и *B* (см. рис. 2.2) управляют процессом передачи данных путем обмена **сообщениями**. Чтобы приложения могли «понимать» получаемую друг от друга информацию, программисты, разрабатывавшие эти приложения, должны *строго оговорить* форматы и последовательность сообщений, которыми приложения будут обмениваться во время выполнения этой операции. Например, они могут договориться о том, что любая операция обмена данными начинается с передачи сообщения, запрашивающего информацию о готовности приложения *B*; что в следующем сообщении идут идентификаторы компьютера и пользователя, сделавшего запрос; что признаком срочного завершения операции обмена данными является определенная кодовая комбинация и т. п. Тем самым определяется **протокол взаимодействия приложений** для выполнения операции данного типа.

Аналогично тому, как при выводе данных на печать необходимо передавать принтеру дополнительно некоторый объем служебной информации — в виде команд управления принтером, так и здесь: для передачи данных из одного компьютера в другой необходимо сопровождать эти данные дополнительной информацией в виде протокольных сообщений, которыми обмениваются приложения.

Заметим, что для реализации протокола нужно, чтобы к моменту возникновения потребности в обмене данными были активны оба приложения: как приложение *A*, которое посылает инициирующее сообщение, так и приложение *B*, которое должно быть готово принять это сообщение и выработать реакцию на него.

Передача любых данных (как сообщений протокола приложений, так и собственно данных, составляющих цель операции обмена) происходит в соответствие с одной и той же процедурой.

На стороне компьютера A приложение, следуя логике протокола, размещает в буфере ОП либо собственное очередное сообщение, либо данные, и обращается к ОС с запросом на выполнение операции межкомпьютерного обмена данными. ОС запускает соответствующий драйвер сетевой карты, который загружает байт из буфера ОП в буфер ИК, после чего инициирует работу ИК. Сетевая интерфейсная карта последовательно передает биты в линию связи, дополняя каждый новый байт стартовым и стоповым битами.

На стороне компьютера B сетевая ИК принимает биты, поступающие со стороны внешнего интерфейса, и помещает их в собственный буфер. После того как получен стоповый бит, интерфейсная карта устанавливает признак завершения приема байта и выполняет проверку корректности приема, например, путем контроля бита четности. Факт корректного приема байта фиксируется драйвером сетевой ИК компьютера *B*. Драйвер переписывает принятый байт из буфера ИК в заранее зарезервированный буфер ОП компьютера *B*.

Приложение *B* извлекает данные из буфера и интерпретирует их в соответствии со своим протоколом либо как сообщение, либо как данные. Если согласно протоколу приложение *B* должно передать ответ приложению *A*, то выполняется симметричная процедура.

Таким образом, связав электрически и информационно два автономно работающих компьютера, мы получили простейшую **компьютерную сеть**.

Доступ к ПУ через сеть

Итак, мы имеем в своем распоряжении механизм, который позволяет приложениям, выполняющимся на разных компьютерах, обмениваться данными. И хотя приложение *A* (см. рис. 2.2) по-прежнему не может управлять принтером, подключенным к компьютеру *B*, оно может теперь воспользоваться средствами межкомпьютерного обмена данными, чтобы передать приложению *B* «просьбу» выполнить для него требуемую операцию. Приложение *A* должно «объяснить» приложению *B*, какую операцию необходимо выполнить, с какими данными, на каком из имеющихся в его распоряжении устройств, в каком виде должен быть распечатан текст и т. п. В ходе печати могут возникнуть ситуации, о которых приложение *B* должно оповестить приложение *A*, например об отсутствии бумаги в принтере. То есть для решения поставленной задачи — доступа к принтеру по сети — должен быть разработан специальный протокол взаимодействия приложений *A* и *B*.

А теперь посмотрим, как работают вместе все элементы этой простейшей компьютерной сети при решении задачи совместного использования принтера.

1. В соответствии с принятым протоколом приложение *A* формирует сообщение-запрос к приложению *B*, помещает его в буфер ОП компьютера *A* и обращается к ОС, снабжая ее необходимой информацией.
2. ОС запускает драйвер сетевой ИК, сообщая ему адрес буфера ОП, где хранится сообщение.
3. Драйвер и сетевая интерфейсная карта компьютера *A*, взаимодействуя с драйвером и интерфейсной картой компьютера *B*, передают сообщение байт за байтом в буфер ОП компьютера *B*.
4. Приложение *B* извлекает сообщение из буфера, интерпретирует его в соответствии с протоколом и выполняет необходимые действия. В число таких действий входит, в том числе, обращение к ОС с запросом на выполнение тех или иных операций с локальным принтером.
5. ОС запускает драйвер принтера, который в кооперации с интерфейсной картой и контроллером принтера выполняет требуемую операцию печати.

Уже на этом начальном этапе, рассматривая связь компьютера с периферийным устройством, мы столкнулись с важнейшими «сетевыми» понятиями: интерфейсом и протоколом, драйвером и интерфейсной картой, а также с проблемами, характерными для компьютерных сетей: согласованием интерфейсов, синхронизацией асинхронных процессов, обеспечением достоверности передачи данных.

Сетевое программное обеспечение

Мы только что рассмотрели случай совместного использования принтера в простейшей сети, состоящей только из двух компьютеров. Однако даже на этом начальном этапе мы

уже можем сделать некоторые выводы относительно строения сетевого программного обеспечения: сетевых служб, сетевой операционной системы и сетевых приложений.

Сетевые службы и сервисы

Потребность в доступе к удаленному принтеру может возникать у пользователей самых разных приложений: текстового редактора, графического редактора, системы управления базой данных (СУБД). Очевидно, что дублирование в каждом из приложений общих для всех них функций по организации удаленной печати является избыточным.

Более эффективным представляется подход, при котором эти функции исключаются из приложений и оформляются в виде пары специализированных программных модулей — *клиента* и *сервера печати* (рис. 2.3), функции которых ранее выполнялись соответственно приложениями *A* и *B*. Теперь эта пара клиент—сервер может быть использована любым приложением, выполняемым на компьютере *A*.

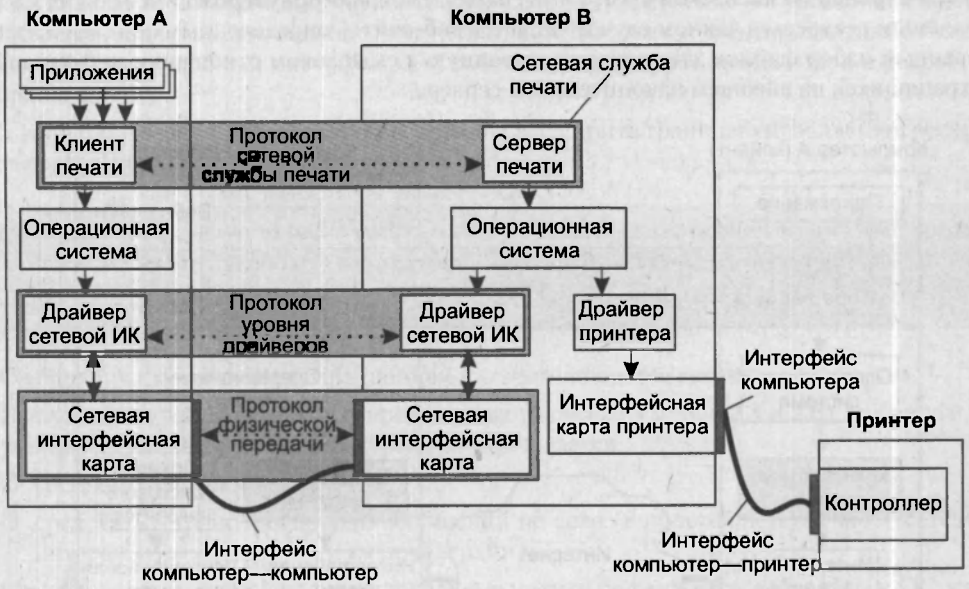


Рис. 2.3. Совместное использование принтера в компьютерной сети с помощью сетевой службы печати

Обобщая такой подход применительно к другим типам разделяемых ресурсов, дадим следующие определения[1]:

Клиент — это модуль, предназначенный для формирования и передачи сообщений-запросов к ресурсам удаленного компьютера от разных приложений с последующим приемом результатов из сети и передачей их соответствующим приложениям.

[1] Термины «клиент» и «сервер» являются чрезвычайно многозначными. Данная пара терминов, уже используемая нами для обозначения функциональной роли взаимодействующих компьютеров и приложений, применима также к программным модулям.

Сервер — это модуль, который постоянно ожидает прихода из сети запросов от клиентов, и приняв запрос, пытается его обслужить, как правило, с участием локальной ОС; один сервер может обслуживать запросы сразу нескольких клиентов (поочередно или одновременно).

Пара клиент—сервер, предоставляющая доступ к конкретному типу ресурса компьютера через сеть, образует **сетевую службу**.

Каждая служба связана с определенным типом сетевых ресурсов. Так, на рис. 2.3 модули клиента и сервера, реализующие удаленный доступ к принтеру, образуют сетевую **службу печати**.

Файловая служба позволяет получать доступ к файлам, хранящимся на диске других компьютеров. Серверный компонент файловой службы называют **файл-сервером**.

Для поиска и просмотра информации в Интернете используется **веб-служба**, состоящая из **веб-сервера** и клиентской программы, называемой **веб-браузером** (web browser). Разделяемым ресурсом в данном случае является **веб-сайт** — определенным образом организованный набор файлов, содержащих связанную в смысловом отношении информацию и хранящихся на внешнем накопителе веб-сервера.

Рис. 2.4. Веб-служба

На схеме веб-службы, показанной на рис. 2.4, два компьютера связаны не непосредственно, как это было во всех предыдущих примерах, а через множество промежуточных компьютеров и других сетевых устройств, входящих в состав Интернета. Для того чтобы отразить

этот факт графически, мы поместили между двумя компьютерами так называемое **коммуникационное облако**, которое позволяет нам абстрагироваться от всех деталей среды передачи сообщений. Обмен сообщениями между клиентской и серверной частями веб-службы выполняется по стандартному протоколу HTTP и никак не зависит от того, передаются ли эти сообщения «из рук в руки» (от интерфейса одного компьютера к интерфейсу другого) или через большое число посредников — транзитных коммуникационных устройств. Вместе с тем, усложнение среды передачи сообщений приводит к возникновению новых дополнительных задач, на решение которых не был рассчитан упоминавшийся ранее простейший драйвер сетевой интерфейсной карты. Вместо него на взаимодействующих компьютерах должны быть установлены более развитые программные **транспортные средства**.

Сетевая операционная система

Операционную систему компьютера часто определяют как взаимосвязанный набор системных программ, который обеспечивает эффективное управление ресурсами компьютера (памятью, процессором, внешними устройствами, файлами и др.), а также предоставляет пользователю удобный интерфейс для работы с аппаратурой компьютера и разработки приложений.

Говоря о *сетевой* ОС, мы, очевидно, должны расширить границы управляемых ресурсов за пределы одного компьютера.

Сетевой операционной системой называют операционную систему компьютера, которая помимо управления локальными ресурсами предоставляет пользователям и приложениям возможность эффективного и удобного доступа к информационным и аппаратным ресурсам других компьютеров сети.

Сегодня практически все операционные системы являются сетевыми.

Из примеров, рассмотренных в предыдущих разделах (см. рис 2.3 и 2.4), мы видим, что удаленный доступ к сетевым ресурсам обеспечивается:

- сетевыми службами;
- средствами транспортировки сообщений по сети (в простейшем случае — сетевыми интерфейсными картами и их драйверами).

Следовательно, именно эти функциональные модули должны быть добавлены к ОС, чтобы она могла называться сетевой (рис. 2.5).

Среди сетевых служб можно выделить такие, которые ориентированы не на простого пользователя, как, например, файловая служба или служба печати, а на администратора. Такие службы направлены на организацию работы сети. Например, **централизованная справочная служба**, или **служба каталогов**, предназначена для ведения базы данных о пользователях сети, обо всех ее программных и аппаратных компонентах[1]. В качестве других примеров можно назвать **службу мониторинга сети**, позволяющую захватывать и анализировать сетевой трафик, **службу безопасности**, в функции которой может входить, в частности, выполнение процедуры логического входа с проверкой пароля, **службу резервного копирования** и **архивирования**.

[1] Например, служба каталогов Active Directory компании Microsoft.

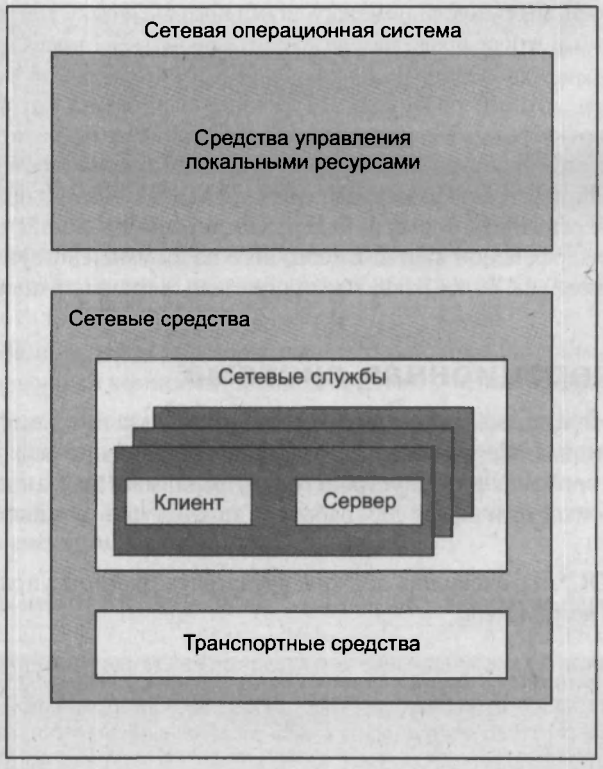


Рис. 2.5. Функциональные компоненты сетевой ОС

От того, насколько богатый набор сетевых служб и услуг предлагает операционная система конечным пользователям, приложениям и администраторам сети, зависит ее позиция в общем ряду сетевых ОС.

Помимо сетевых служб сетевая ОС должна включать *программные коммуникационные (транспортные) средства*, обеспечивающие совместно с аппаратными коммуникационными средствами передачу сообщений, которыми обмениваются клиентские и серверные части сетевых служб. Задачу коммуникации между компьютерами сети решают драйверы и протокольные модули. Они выполняют такие функции, как формирование сообщений, разбиение сообщения на части (пакеты, кадры), преобразование имен компьютеров в числовые адреса, дублирование сообщений в случае их потери, определение маршрута в сложной сети и т. д.

И сетевые службы, и транспортные средства могут являться неотъемлемыми (встроенными) компонентами ОС или существовать в виде отдельных программных продуктов. Например, сетевая файловая служба обычно встраивается в ОС, а вот веб-браузер чаще всего приобретается отдельно. Типичная сетевая ОС имеет в своем составе широкий набор драйверов и протокольных модулей, однако у пользователя, как правило, есть возможность дополнить этот стандартный набор необходимыми ему программами. Решение о способе реализации клиентов и серверов сетевой службы, а также драйверов и протокольных модулей принимается разработчиками с учетом самых разных соображений: технических, ком-

мерческих и даже юридических. Так, например, именно на основании антимонопольного закона США компании Microsoft было запрещено включать ее браузер Internet Explorer в состав ОС этой компании.

Сетевая служба может быть представлена в ОС либо обеими (клиентской и серверной) частями, либо только одной из них.

В первом случае операционная система, называемая **одноранговой**, не только позволяет обращаться к ресурсам других компьютеров, но и предоставляет собственные ресурсы в распоряжение пользователей других компьютеров. Например, если на всех компьютерах сети установлены и клиенты, и серверы файловой службы, то все пользователи сети могут совместно применять файлы друг друга. Компьютеры, совмещающие функции клиента и сервера, называют одноранговыми узлами.

Операционная система, которая преимущественно содержит клиентские части сетевых служб, называется **клиентской**. Клиентские ОС устанавливаются на компьютеры, обращающиеся с запросами к ресурсам других компьютеров сети. За такими компьютерами, также называемыми клиентскими, работают рядовые пользователи. Обычно клиентские компьютеры относятся к классу относительно простых устройств.

К другому типу операционных систем относится **серверная ОС** — она ориентирована на обработку запросов из сети к ресурсам своего компьютера и включает в себя в основном серверные части сетевых служб. Компьютер с установленной на нем серверной ОС, занимающийся исключительно обслуживанием запросов других компьютеров, называют **выделенным сервером** сети. За выделенным сервером, как правило, обычные пользователи не работают.

ПРИМЕЧАНИЕ

Подробнее о сетевых операционных системах и встроенных в них сетевых службах вы можете прочитать в специальной литературе, а также в учебнике авторов «Сетевые операционные системы». Наиболее популярные сетевые службы Интернета, такие как электронная почта, веб-служба, IP-телефония и др., рассматриваются в главе 23.

Сетевые приложения

Компьютер, подключенный к сети, может выполнять следующие типы приложений:

- **Локальное приложение** целиком выполняется на данном компьютере и использует только локальные ресурсы (рис. 2.6, *а*). Для такого приложения не требуется никаких сетевых средств, оно может быть выполнено на автономно работающем компьютере.
- **Централизованное сетевое приложение** целиком выполняется на данном компьютере, но обращается в процессе своего выполнения к ресурсам других компьютеров сети. В примере на рисунке 2.6, *б* приложение, которое выполняется на клиентском компьютере, обрабатывает данные из файла, хранящегося на файл-сервере, а затем распечатывает результаты на принтере, подключенном к серверу печати. Очевидно, что работа такого типа приложений невозможна без участия сетевых служб и средств транспортировки сообщений.
- **Распределенное (сетевое) приложение** состоит из нескольких взаимодействующих частей, каждая из которых выполняет какую-то определенную законченную работу

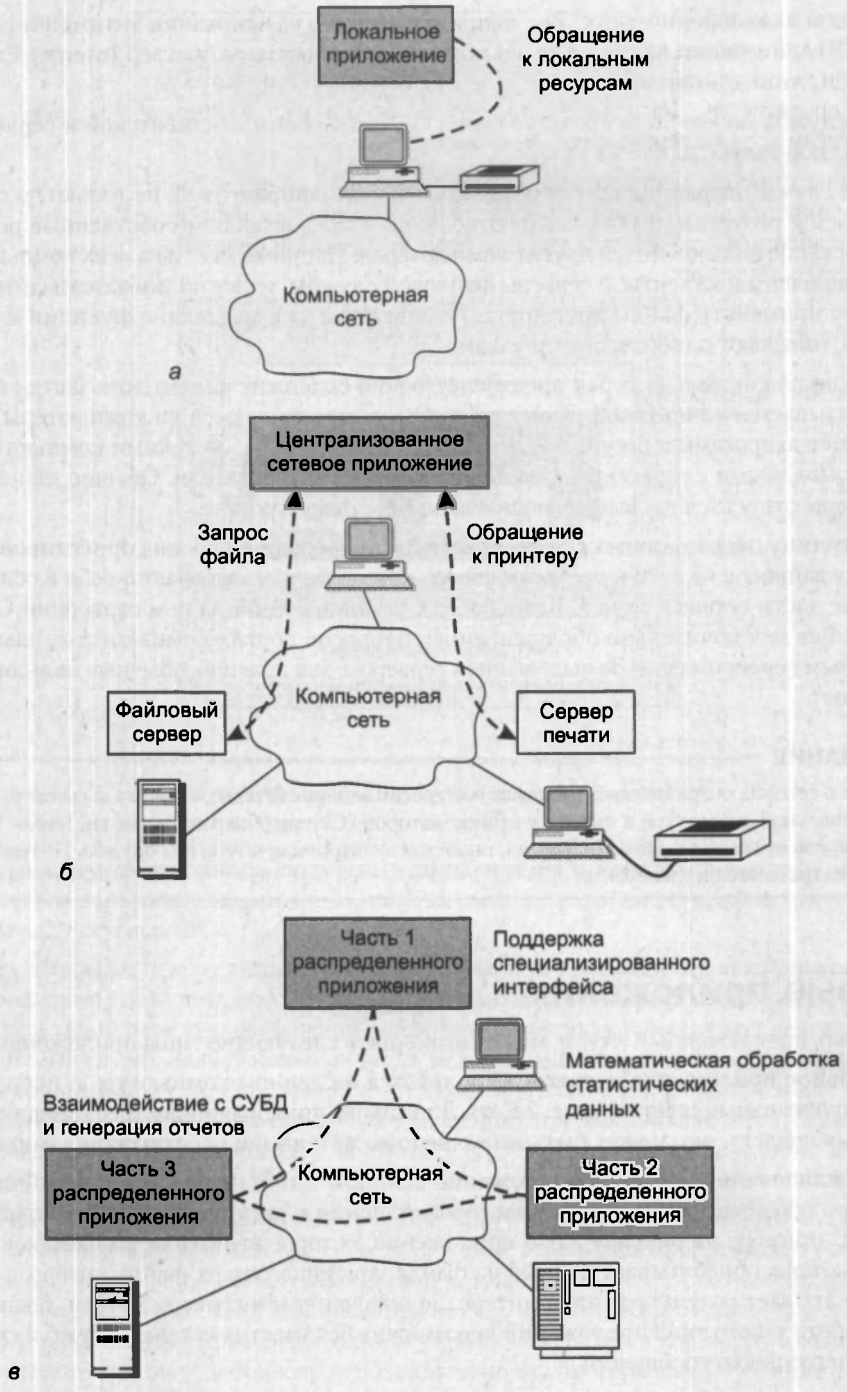


Рис. 2.6. Типы приложений, выполняющихся в сети

по решению прикладной задачи, причем каждая часть может выполняться и, как правило, выполняется на отдельном компьютере сети (рис. 2.6, *в*). Части распределенного приложения взаимодействуют друг с другом, используя сетевые службы и транспортные средства ОС. Распределенное приложение в общем случае имеет доступ ко всем ресурсам компьютерной сети.

Очевидным преимуществом распределенных приложений является возможность распараллеливания вычислений, а также специализация компьютеров. Так, в приложении, предназначенном, скажем, для анализа климатических изменений, можно выделить три достаточно самостоятельные части (см. рис. 2.6, *в*), допускающие распараллеливание. Первая часть приложения, выполняющаяся на сравнительно маломощном персональном компьютере, могла бы поддерживать специализированный графический пользовательский интерфейс, вторая — заниматься статистической обработкой данных на высокопроизводительном мэйнфрейме, а третья — генерировать отчеты на сервере с установленной стандартной СУБД. В общем случае каждая из частей распределенного приложения может быть представлена несколькими копиями, работающими на разных компьютерах. Скажем, в данном примере часть 1, ответственную за поддержку специализированного пользовательского интерфейса, можно было бы запустить на нескольких персональных компьютерах, что позволило бы работать с этим приложением нескольким пользователям одновременно.

Однако чтобы добиться всех тех преимуществ, которые сулят распределенные приложения, разработчикам этих приложений приходится решать множество проблем, например: на сколько частей следует разбить приложение, какие функции возложить на каждую часть, как организовать взаимодействие этих частей, чтобы в случае сбоев и отказов оставшиеся части корректно завершали работу и т. д., и т. п.

Заметим, что все сетевые службы, включая файловую службу, службу печати, службу электронной почты, службу удаленного доступа, интернет-телефонию и т. д., по определению относятся к классу распределенных приложений. Действительно, любая сетевая служба включает в себя клиентскую и серверную части, которые могут и обычно выполняются на разных компьютерах.

На рис. 2.7, иллюстрирующем распределенный характер веб-службы, мы видим различные виды клиентских устройств — персональные компьютеры, ноутбуки и мобильные телефоны — с установленными на них веб-браузерами, которые взаимодействуют по сети с веб-сервером. Таким образом, с одним и тем же веб-сайтом может одновременно работать множество — сотни и тысячи — сетевых пользователей.

Многочисленные примеры распределенных приложений можно встретить и в такой области, как обработка данных научных экспериментов. Это не удивительно, так как многие эксперименты порождают такие большие объемы данных, генерируемых в реальном масштабе времени, которые просто невозможно обработать на одном, даже очень мощном, суперкомпьютере. Кроме того, алгоритмы обработки экспериментальных данных часто легко распараллеливаются, что также важно для успешного применения взаимосвязанных компьютеров с целью решения какой-либо общей задачи. Одним из последних и очень известных примеров распределенного научного приложения является программное обеспечение обработки данных большого адронного коллайдера (Large Hadron Collider, LHC), запущенного 10 сентября 2008 года в CERN — это приложение работает более чем на 30 тысячах компьютеров, объединенных в сеть.

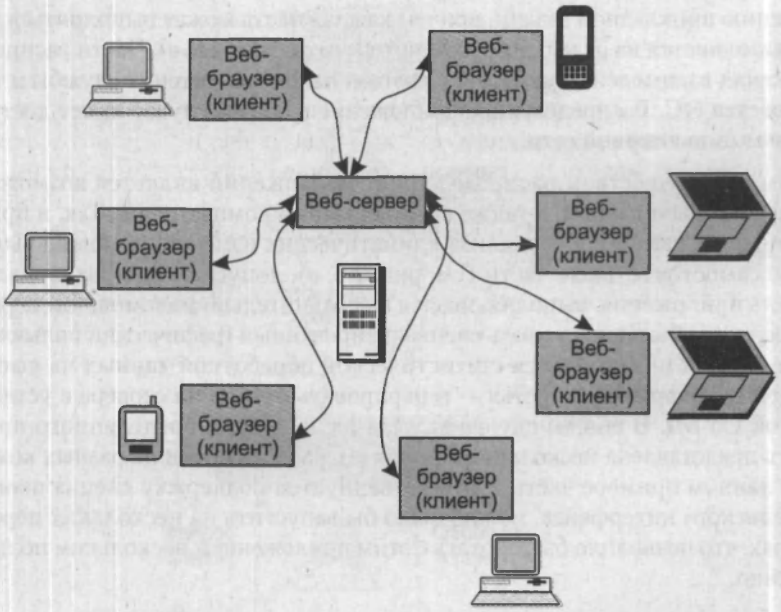


Рис. 2.7. Веб-служба как распределенное приложение

Физическая передача данных по линиям связи

Даже при рассмотрении простейшей сети, состоящей всего из двух машин, можно выявить многие проблемы, связанные с физической передачей сигналов по линиям связи.

Кодирование

В вычислительной технике для представления данных используется **двоичный код**. Внутри компьютера единицам и нулям данных соответствуют дискретные электрические сигналы.

Представление данных в виде электрических или оптических сигналов называется **кодированием**.

Существуют различные способы кодирования двоичных цифр, например **потенциальный способ**, при котором единице соответствует один уровень напряжения, а нулю — другой, или **импульсный способ**, когда для представления цифр используются импульсы различной полярности.

Аналогичные подходы применимы для кодирования данных и при передаче их между двумя компьютерами **по линиям связи**. Однако эти линии связи отличаются по своим характеристикам от линий внутри компьютера. Главное отличие внешних линий связи от внутренних состоит в их гораздо большей протяженности, а также в том, что они проходят

вне экранированного корпуса по пространствам, зачастую подверженным воздействию сильных электромагнитных помех. Все это приводит к существенно большим искажениям прямоугольных импульсов (например, «заваливанию» фронтов), чем внутри компьютера. Поэтому для надежного распознавания импульсов на приемном конце линии связи при передаче данных внутри и вне компьютера не всегда можно использовать одни и те же скорости и способы кодирования. Например, медленное нарастание фронта импульса из-за высокой емкостной нагрузки линии требует, чтобы импульсы передавались с меньшей скоростью (чтобы передний и задний фронты соседних импульсов не перекрывались, и импульс успел «дорасти» до требуемого уровня).

В вычислительных сетях применяют как потенциальное, так и импульсное кодирование дискретных данных, а также специфический способ представления данных, который никогда не используется внутри компьютера, — **модуляцию** (рис. 2.8). При модуляции дискретная информация представляется синусоидальным сигналом той частоты, которую хорошо передает имеющаяся линия связи.

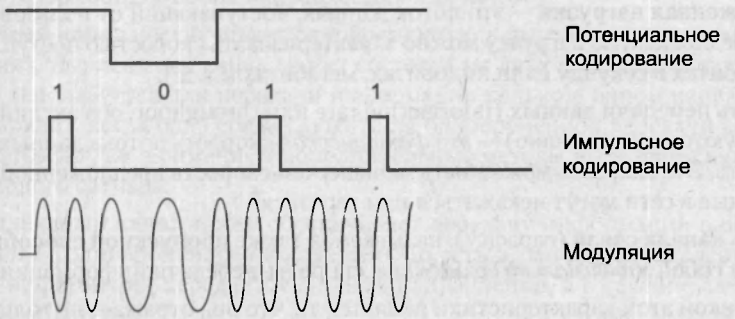


Рис. 2.8. Примеры представления дискретной информации

Потенциальное или импульсное кодирование применяется на каналах *высокого качества*, а модуляция на основе синусоидальных сигналов предпочтительнее в том случае, когда канал вносит сильные искажения в передаваемые сигналы. Например, модуляция используется в глобальных сетях при передаче данных через аналоговые телефонные каналы связи, которые были разработаны для передачи голоса в аналоговой форме и поэтому плохо подходят для непосредственной передачи импульсов.

На способ передачи сигналов влияет и *количество проводов* в линиях связи между компьютерами. Для снижения стоимости линий связи в сетях обычно стремятся к сокращению количества проводов и из-за этого передают все биты одного байта или даже нескольких байтов не параллельно, как это делается внутри компьютера, а последовательно (побитно), для чего достаточно всего одной пары проводов.

Еще одной проблемой, которую нужно решать при передаче сигналов, является проблема взаимной **синхронизации** передатчика одного компьютера с приемником другого. При организации взаимодействия модулей внутри компьютера эта проблема решается очень просто, так как в этом случае все модули синхронизируются от общего тактового генератора. Проблема синхронизации при связи компьютеров может решаться разными способами, как путем обмена специальными тактовыми синхроимпульсами по отдельной линии, так и путем периодической синхронизации заранее обусловленными кодами или импульсами характерной формы, отличающейся от формы импульсов данных.

Несмотря на предпринимаемые меры (выбор соответствующей скорости обмена данными, линий связи с определенными характеристиками, способа синхронизации приемника и передатчика), существует вероятность искажения некоторых битов передаваемых данных. Для повышения надежности передачи данных между компьютерами, как правило, используется стандартный прием — подсчет **контрольной суммы** и передача полученного значения по линиям связи после каждого байта или после некоторого блока байтов. Часто в протокол обмена данными включается как обязательный элемент **сигнал-квитанция**, который подтверждает правильность приема данных и посылается от получателя отправителю.

Характеристики физических каналов

Существует большое количество характеристик, связанных с передачей трафика через физические каналы. С теми из них, которые будут необходимы нам уже в ближайшее время, мы познакомимся сейчас, а некоторые изучим позже, в главе 6.

- **Предложенная нагрузка** — это поток данных, поступающий от пользователя на вход сети. Предложенную нагрузку можно характеризовать скоростью поступления данных в сеть в битах в секунду (или килобитах, мегабитах и т. д.).
- **Скорость передачи данных** (information rate или throughput, оба английских термина используются равноправно) — это *фактическая* скорость потока данных, прошедшего через сеть. Эта скорость может быть меньше, чем скорость предложенной нагрузки, так как данные в сети могут искажаться или теряться.
- **Емкость канала связи** (capacity), называемая также **пропускной способностью**, представляет собой *максимально возможную* скорость передачи информации по каналу.
- Спецификой этой характеристики является то, что она отражает не только параметры *физической среды передачи*, но и особенности *выбранного способа передачи* дискретной информации по этой среде. Например, емкость канала связи в сети Ethernet на оптическом волокне равна 10 Мбит/с. Эта скорость является предельно возможной для сочетания технологии Ethernet и оптического волокна. Однако для того же самого оптического волокна можно разработать другую технологию передачи данных, отличающуюся способом кодирования данных, тактовой частотой и другими параметрами, которая будет иметь другую емкость. Так, технология Fast Ethernet обеспечивает передачу данных по тому же оптическому волокну с максимальной скоростью 100 Мбит/с, а технология Gigabit Ethernet — 1000 Мбит/с. Передатчик коммуникационного устройства должен работать со скоростью, равной пропускной способности канала. Эта скорость иногда называется **битовой скоростью передатчика** (bit rate of transmitter).
- **Полоса пропускания** (bandwidth) — этот термин может ввести в заблуждение, потому что он используется в двух разных значениях. Во-первых, с его помощью могут характеризовать *среду передачи*. В этом случае он означает ширину полосы частот, которую линия передает без существенных искажений. Из этого определения понятно происхождение термина. Во-вторых, термин «полоса пропускания» используется как синоним термина *емкость канала связи*. В первом случае полоса пропускания измеряется в герцах (Гц), во втором — в битах в секунду. Различать значения термина нужно по контексту, хотя иногда это достаточно трудно. Конечно, лучше было бы применять разные термины для различных характеристик, но существуют традиции, которые изменить трудно. Такое двойное использование термина «полоса пропускания» уже

вошло во многие стандарты и книги, поэтому и в данной книге мы будем следовать сложившемуся подходу. Нужно также учитывать, что этот термин в его втором значении является даже более распространенным, чем емкость, поэтому из этих двух синонимов мы будем использовать полосу пропускания.

Еще одна группа характеристик канала связи связана с возможностью передачи информации по каналу в одну или обе стороны.

При взаимодействии двух компьютеров обычно требуется передавать информацию в обоих направлениях, от компьютера *A* к компьютеру *B* и обратно. Даже в том случае, когда пользователю кажется, что он только получает информацию (например, загружает музыкальный файл из Интернета) или только ее передает (отправляет электронное письмо), обмен информации идет в двух направлениях. Просто существует основной поток данных, которые интересуют пользователя, и вспомогательный поток противоположного направления, который образуют квитанции о получении этих данных.

Физические каналы связи делятся на несколько типов в зависимости от того, могут они передавать информацию в обоих направлениях или нет.

- **Дуплексный канал** обеспечивает одновременную передачу информации в обоих направлениях. Дуплексный канал может состоять их двух физических сред, каждая их которых используется для передачи информации только в одном направлении. Возможен вариант, когда одна среда служит для одновременной передачи встречных потоков, в этом случае применяют дополнительные методы выделения каждого потока из суммарного сигнала.
- **Полудуплексный канал** также обеспечивает передачу информации в обоих направлениях, но не одновременно, а по очереди. То есть в течение определенного периода времени информация передается в одном направлении, а в течение следующего периода — в обратном.
- **Симплексный канал** позволяет передавать информацию только в одном направлении. Часто дуплексный канал состоит из двух симплексных каналов.

Подробно вопросы физической передачи дискретных данных обсуждаются в части II.

Проблемы связи нескольких компьютеров

До сих пор мы рассматривали вырожденную сеть, состоящую всего из двух машин. При объединении в сеть большего числа компьютеров возникает целый комплекс новых проблем.

Топология физических связей

Объединяя в сеть несколько (больше двух) компьютеров, необходимо решить, каким образом соединить их друг с другом, другими словами, выбрать конфигурацию физических связей, или топологию.

Под **топологией сети** понимается конфигурация графа, вершинам которого соответствуют конечные узлы сети (например, компьютеры) и коммуникационное оборудование (например, маршрутизаторы), а ребрам — физические или информационные связи между вершинами.

Число возможных вариантов конфигурации резко возрастает при увеличении числа связываемых устройств. Так, если три компьютера мы можем связать двумя способами (рис. 2.9, *а*), то для четырех можно предложить уже шесть топологически разных конфигураций (при условии неразличимости компьютеров), что и иллюстрирует рис. 2.9, *б*.

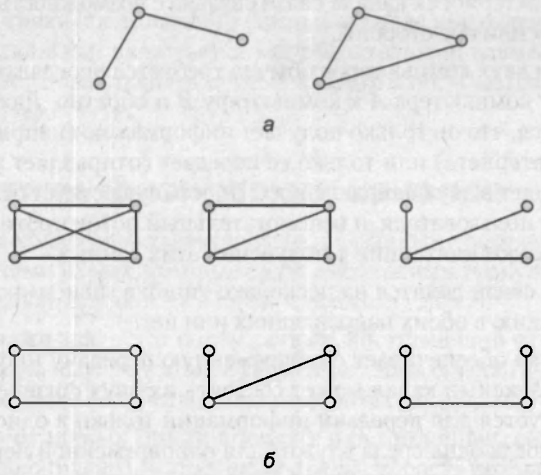


Рис. 2.9. Варианты связи компьютеров

Мы можем соединять каждый компьютер с каждым или же связывать их последовательно, предполагая, что они будут общаться, передавая сообщения друг другу «транзитом». Транзитные узлы должны быть оснащены специальными средствами, позволяющими им выполнять эту специфическую посредническую операцию. В качестве транзитного узла может выступать как универсальный компьютер, так и специализированное устройство.

От выбора топологии связей существенно зависят характеристики сети. Например, наличие между узлами нескольких путей повышает надежность сети и делает возможным распределение загрузки между отдельными каналами. Простота присоединения новых узлов, свойственная некоторым топологиям, делает сеть легко *расширяемой*. Экономические соображения часто приводят к выбору топологий, для которых характерна минимальная суммарная длина линий связи.

Среди множества возможных конфигураций различают полносвязные и неполносвязные.

Полносвязная топология (рис. 2.10, *а*) соответствует сети, в которой каждый компьютер непосредственно связан со всеми остальными. Несмотря на логическую простоту, этот вариант оказывается громоздким и неэффективным. Действительно, в таком случае каждый компьютер в сети должен иметь большое количество коммуникационных портов, достаточное для связи с каждым из остальных компьютеров сети. Для каждой пары компьютеров должна быть выделена отдельная физическая линия связи. (В некоторых случаях даже две, если невозможно использование этой линии для двусторонней передачи.) Полносвязные топологии в крупных сетях применяются редко, так как для связи N узлов требуется $N(N-1)/2$ физических дуплексных линий связей, то есть имеет место квадратичная зависимость от числа узлов. Чаще этот вид топологии используется в многомашинных комплексах или в сетях, объединяющих небольшое количество компьютеров.

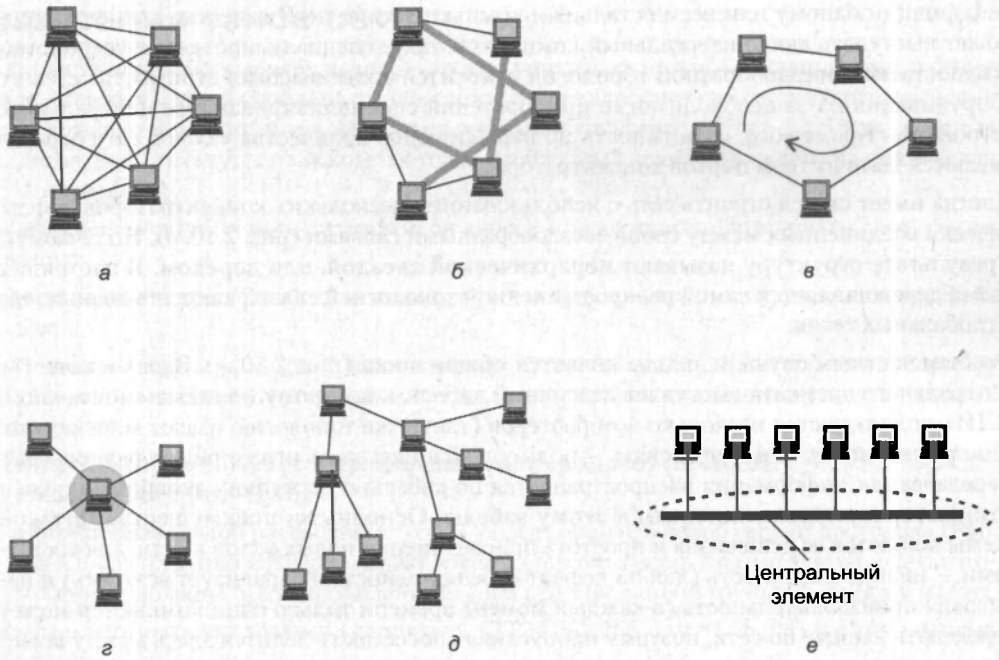


Рис. 2.10. Типовые топологии сетей

Все другие варианты основаны на **неполносвязных топологиях**, когда для обмена данными между двумя компьютерами может потребоваться транзитная передача данных через другие узлы сети.

Ячеистая топология[1] получается из полносвязной путем удаления некоторых связей (рис. 2.10, *б*). Ячеистая топология допускает соединение большого количества компьютеров и характерна, как правило, для крупных сетей.

В сетях с **кольцевой топологией** (рис. 2.10, *в*) данные передаются по кольцу от одного компьютера к другому. Главным достоинством кольца является то, что оно по своей природе обеспечивает резервирование связей. Действительно, любая пара узлов соединена здесь двумя путями — по часовой стрелке и против нее. Кроме того, кольцо представляет собой очень удобную конфигурацию для организации обратной связи — данные, сделав полный оборот, возвращаются к узлу-источнику. Поэтому источник может контролировать процесс доставки данных адресату. Часто это свойство кольца используется для тестирования связности сети и поиска узла, работающего некорректно. В то же время в сетях с кольцевой топологией необходимо принимать специальные меры, чтобы в случае выхода из строя или отключения какого-либо компьютера не прерывался канал связи между остальными узлами кольца.

Звездообразная топология (рис. 2.10, *г*) образуется в случае, когда каждый компьютер подключается непосредственно к общему центральному устройству, называемому **концентратором**[2]. В функции концентратора входит направление передаваемой компьютером

[1] Иногда ячеистой называют полносвязную или близкую к полносвязной топологию.

[2] В данном случае термин «концентратор» используется в широком смысле, обозначая любое многовходовое устройство, способное служить центральным элементом, например коммутатор или маршрутизатор.

информации одному или всем остальным компьютерам сети. В качестве концентратора может выступать как универсальный компьютер, так и специализированное устройство. К недостаткам звездообразной топологии относится более высокая стоимость сетевого оборудования из-за необходимости приобретения специализированного центрального устройства. Кроме того, возможности по наращиванию количества узлов в сети ограничиваются количеством портов концентратора.

Иногда имеет смысл строить сеть с использованием нескольких концентраторов, иерархически соединенных между собой звездообразными связями (рис. 2.10, *д*). Получаемую в результате структуру называют и**ерархической звездой**, или **деревом**. В настоящее время дерево является самой распространенной топологией связей как в локальных, так и глобальных сетях.

Особым частным случаем звезды является **общая шина** (рис. 2.10, *е*). Здесь в качестве центрального элемента выступает пассивный кабель, к которому по схеме «монтажного ИЛИ» подключается несколько компьютеров (такую же топологию имеют многие сети, использующие беспроводную связь — роль общей шины здесь играет общая радиосреда). Передаваемая информация распространяется по кабелю и доступна одновременно всем компьютерам, присоединенным к этому кабелю. Основными преимуществами такой схемы являются ее дешевизна и простота присоединения новых узлов к сети, а недостатками — низкая надежность (любой дефект кабеля полностью парализует всю сеть) и невысокая производительность (в каждый момент времени только один компьютер может передавать данные по сети, поэтому пропускная способность делится здесь между всеми узлами сети).

В то время как небольшие сети, как правило, имеют типовую топологию — звезда, кольцо или общая шина, для крупных сетей характерно наличие произвольных связей между компьютерами. В таких сетях можно выделить отдельные произвольно связанные фрагменты (подсети), имеющие типовую топологию, поэтому их называют сетями со **смешанной топологией** (рис. 2.11).

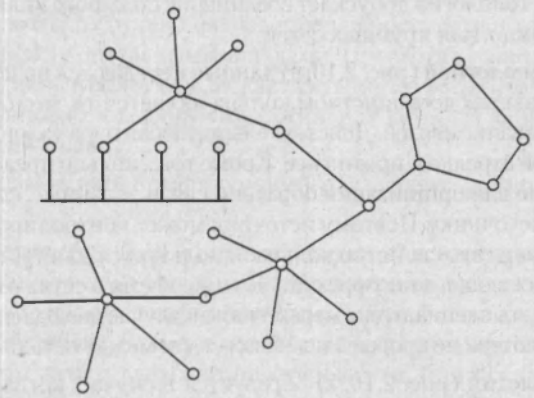


Рис. 2.11. Смешанная топология

Адресация узлов сети

Еще одной новой проблемой, которую нужно учитывать при объединении трех и более компьютеров, является проблема их адресации, точнее адресации их сетевых интерфейсов[1]. Один компьютер может иметь несколько сетевых интерфейсов. Например, для создания полносвязной структуры из N компьютеров необходимо, чтобы у каждого из них имелся $N - 1$ интерфейс.

По количеству адресуемых интерфейсов адреса можно классифицировать следующим образом:

- **уникальный адрес** (unicast) используется для идентификации отдельных интерфейсов;
- **групповой адрес** (multicast) идентифицирует сразу несколько интерфейсов, поэтому данные, помеченные групповым адресом, доставляются каждому из узлов, входящих в группу;
- данные, направленные по **широковещательному адресу** (broadcast), должны быть доставлены всем узлам сети;
- **адрес произвольной рассылки** (anycast), определенный в новой версии протокола IPv6, так же, как и групповой адрес, задает группу адресов, однако данные, посланные по этому адресу, должны быть доставлены не всем адресам данной группы, а любому из них.

Адреса могут быть **числовыми** (например, 129.26.255.255 или 81.1a.ff.ff) и **символьными** (site.domen.ru, willi-winki).

Символьные адреса (имена) предназначены для запоминания людьми и поэтому обычно несут смысловую нагрузку. Для работы в больших сетях символьное имя может иметь иерархическую структуру, например ftp-arch1.ucl.ac.uk. Этот адрес говорит о том, что данный компьютер поддерживает ftp-архив в сети одного из колледжей Лондонского университета (University College London — ucl) и эта сеть относится к академической ветви (ac) Интернета Великобритании (United Kingdom — uk). При работе в пределах сети Лондонского университета такое длинное символьное имя явно избыточно и вместо него можно пользоваться кратким символьным именем ftp-arch1. Хотя символьные имена удобны для людей, из-за переменного формата и потенциально большой длины их передача по сети не очень экономична.

Множество всех адресов, которые являются допустимыми в рамках некоторой схемы адресации, называется **адресным пространством**.

Адресное пространство может иметь плоскую (линейную) организацию (рис. 2.12) или иерархическую организацию (рис. 2.13).

При **плоской организации** множество адресов никак не структурировано. Примером плоского числового адреса является **MAC-адрес**, предназначенный для однозначной идентификации сетевых интерфейсов в локальных сетях. Такой адрес обычно используется только аппаратурой, поэтому его стараются сделать по возможности компактным и записывают в виде двоичного или шестнадцатеричного числа, например 0081005e24a8. При задании

[1] Иногда вместо точного выражения «адрес сетевого интерфейса» мы будем использовать упрощенное — «адрес узла сети».

MAC-адресов не требуется выполнение ручной работы, так как они обычно встраиваются в аппаратуру компанией-изготовителем, поэтому их называют также **аппаратными адресами** (hardware address). Использование плоских адресов является жестким решением — при замене аппаратуры, например сетевого адаптера, изменяется и адрес сетевого интерфейса компьютера.

Рис. 2.12. Плоская организация адресного пространства

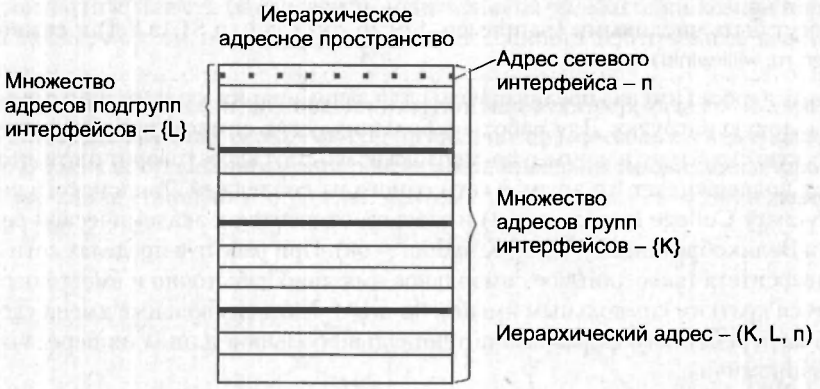


Рис. 2.13. Иерархическая организация адресного пространства

При **иерархической** организации адресное пространство структурируется в виде вложенных друг в друга подгрупп, которые, последовательно сужая адресуемую область, в конце концов, определяют отдельный сетевой интерфейс.

В показанной на рис. 2.13 трехуровневой структуре адресного пространства адрес конечного узла задается тремя составляющими: идентификатором группы (K), в которую входит данный узел, идентификатором подгруппы (L) и, наконец, идентификатором узла (n), однозначно определяющим его в подгруппе. Иерархическая адресация во многих случаях оказывается более рациональной, чем плоская. В больших сетях, состоящих из многих тысяч узлов, использование плоских адресов приводит к большим издержкам — конечным узлам и коммуникационному оборудованию приходится оперировать таблицами адресов, состоящими из тысяч записей. В противоположность этому иерархическая система адресации позволяет при перемещении данных до определенного момента пользоваться

только старшей составляющей адреса (например, идентификатором группы K), затем для дальнейшей локализации адресата задействовать следующую по старшинству часть (L) и в конечном счете — младшую часть (n).

Типичными представителями иерархических числовых адресов являются сетевые IP- и IPX-адреса. В них поддерживается двухуровневая иерархия, адрес делится на старшую часть — номер сети и младшую — номер узла. Такое деление позволяет передавать сообщения между сетями только на основании номера сети, а номер узла требуется уже после доставки сообщения в нужную сеть; точно так же, как название улицы используется почтальоном только после того, как письмо доставлено в нужный город.

На практике обычно применяют сразу несколько схем адресации, так что сетевой интерфейс компьютера может одновременно иметь несколько адресов-имен. Каждый адрес задействуется в той ситуации, когда соответствующий вид адресации наиболее удобен. А для преобразования адресов из одного вида в другой используются специальные вспомогательные протоколы, которые называют **протоколами разрешения адресов**.

Пользователи адресуют компьютеры иерархическими символьными именами, которые автоматически заменяются в сообщениях, передаваемых по сети, иерархическими числовыми адресами. С помощью этих числовых адресов сообщения доставляются из одной сети в другую, а после доставки сообщения в сеть назначения вместо иерархического числового адреса используется плоский аппаратный адрес компьютера. Проблема установления соответствия между адресами различных типов может решаться как централизованными, так и распределенными средствами.

При *централизованном подходе* в сети выделяется один или несколько компьютеров (серверов имен), в которых хранится таблица соответствия имен различных типов, например символьных имен и числовых адресов. Все остальные компьютеры обращаются к серверу имен с запросами, чтобы по символьному имени найти числовой номер необходимого компьютера.

При *распределенном подходе* каждый компьютер сам хранит все назначенные ему адреса разного типа. Тогда компьютер, которому необходимо определить по известному иерархическому числовому адресу некоторого компьютера его плоский аппаратный адрес, посылает в сеть широковещательный запрос. Все компьютеры сети сравнивают содержащийся в запросе адрес с собственным. Тот компьютер, у которого обнаружилось совпадение, посылает ответ, содержащий искомый аппаратный адрес. Такая схема использована в **протоколе разрешения адресов** (Address Resolution Protocol, ARP) стека TCP/IP.

Достоинство распределенного подхода состоит в том, что он позволяет отказаться от выделения специального компьютера в качестве сервера имен, который, к тому же, часто требует ручного задания таблицы соответствия адресов. Недостатком его является необходимость широковещательных сообщений, перегружающих сеть. Именно поэтому распределенный подход используется в небольших сетях, а централизованный — в больших.

До сих пор мы говорили об адресах сетевых интерфейсов, компьютеров и коммуникационных устройств, однако конечной целью данных, пересылаемых по сети, являются не сетевые интерфейсы или компьютеры, а выполняемые на этих устройствах программы — процессы. Поэтому в адресе назначения наряду с информацией, идентифицирующей интерфейс устройства, должен указываться адрес процесса, которому предназначены посылаемые по сети данные. Очевидно, что достаточно обеспечить уникальность адреса процесса в пределах компьютера. Примером адресов процессов являются *номера портов TCP и UDP*, используемые в стеке TCP/IP.

Коммутация

Итак, пусть компьютеры физически связаны между собой в соответствии с некоторой топологией и выбрана система адресации. Остается нерешенной самая важная проблема: каким способом передавать данные между конечными узлами? Особую сложность приобретает эта задача для неполносвязной топологии сети, когда обмен данными между произвольной парой конечных узлов (пользователей) должен идти в общем случае через транзитные узлы.

Соединение конечных узлов через сеть транзитных узлов называют коммутацией. Последовательность узлов, лежащих на пути от отправителя к получателю, образует маршрут.

Например, в сети, показанной на рис. 2.14, узлы *2* и *4*, непосредственно между собой не связанные, вынуждены передавать данные через транзитные узлы, в качестве которых могут выступить, например, узлы *1* и *5*. Узел *1* должен выполнить передачу данных между своими интерфейсами *A* и *B*, а узел *5* — между интерфейсами *F* и *B*. В данном случае маршрутом является последовательность: *2-1-5-4*, где *2* — узел-отправитель, *1* и *5* — транзитные узлы, *4* — узел-получатель.

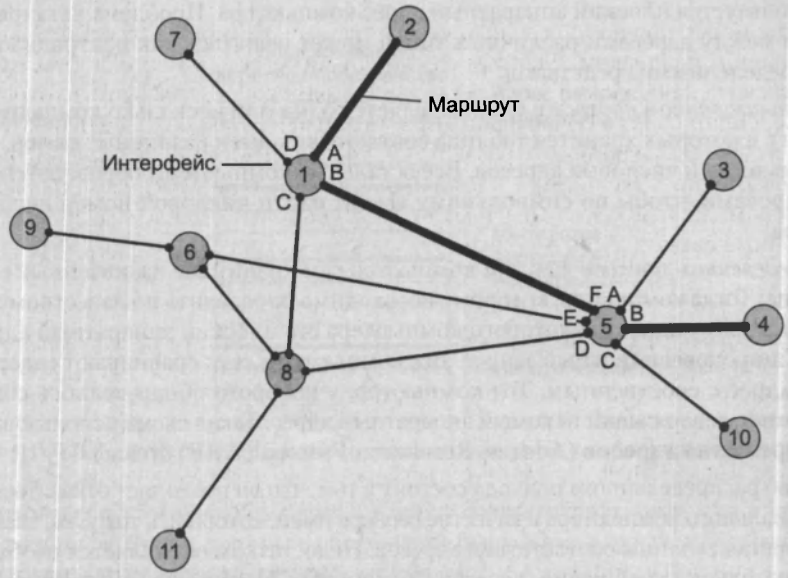


Рис. 2.14. Коммутация абонентов через сеть транзитных узлов

Обобщенная задача коммутации

В самом общем виде задача коммутации может быть представлена в виде следующих взаимосвязанных частных задач.

1. Определение информационных потоков, для которых требуется прокладывать маршруты.

2. Маршрутизация потоков.
3. Продвижение потоков, то есть распознавание потоков и их локальная коммутация на каждом транзитном узле.
4. Мультиплексирование и демультиплексирование потоков.

Определение информационных потоков

Понятно, что через один транзитный узел может проходить несколько маршрутов, например, через узел *5* (см. рис. 2.14) проходят, как минимум, все данные, направляемые узлом *4* каждому из остальных узлов, а также все данные, поступающие в узлы *3*, *4* и *10*. Транзитный узел должен уметь *распознавать* поступающие на него потоки данных, для того чтобы обеспечивать передачу каждого из них именно на тот свой интерфейс, который ведет к нужному узлу.

Информационным потоком, или потоком данных, называют непрерывную последовательность данных, объединенных набором общих признаков, выделяющих эти данные из общего сетевого трафика.

Например, как поток можно определить все данные, поступающие от одного компьютера; объединяющим признаком в данном случае служит адрес источника. Эти же данные можно представить как совокупность нескольких **подпотоков**, каждый из которых в качестве дифференцирующего признака имеет адрес назначения. Наконец, каждый из этих подпотоков, в свою очередь, можно разделить на более мелкие подпотоки, порожденные разными сетевыми приложениями — электронной почтой, программой копирования файлов, веб-сервером. Данные, образующие поток, могут быть представлены в виде различных информационных единиц данных — пакетов, кадров или ячеек.

ПРИМЕЧАНИЕ

В англоязычной литературе для потоков данных, передающихся с равномерной и неравномерной скоростью, обычно используют разные термины — соответственно «data stream» и «data flow». Например, при передаче веб-страницы через Интернет предложенная нагрузка представляет собой неравномерный поток данных, а при вещании музыки интернет-станцией — равномерный. Для сетей передачи данных характерна неравномерная скорость передачи, поэтому далее в большинстве ситуаций под термином «поток данных» мы будем понимать именно неравномерный поток данных и указывать на равномерный характер этого процесса только тогда, когда это нужно подчеркнуть.

Очевидно, что при коммутации в качестве обязательного признака выступает **адрес назначения** данных. На основании этого признака весь поток входящих в транзитный узел данных разделяется на подпотоки, каждый из которых передается на интерфейс, соответствующий маршруту продвижения данных.

Адреса источника и назначения определяют поток для пары соответствующих конечных узлов. Однако часто бывает полезно представить этот поток в виде нескольких подпотоков, причем для каждого из них может быть проложен свой особый маршрут. Рассмотрим пример, когда на одной и той же паре конечных узлов выполняется несколько взаимодействующих по сети приложений, каждое из которых предъявляет к сети свои особые требования. В таком случае выбор маршрута должен осуществляться с учетом характера передаваемых

данных, например, для файлового сервера важно, чтобы передаваемые им большие объемы данных направлялись по каналам, обладающим высокой пропускной способностью, а для программной системы управления, которая посылает в сеть короткие сообщения, требующие обязательной и немедленной отработки, при выборе маршрута более важна надежность линии связи и минимальный уровень задержек на маршруте. Кроме того, даже для данных, предъявляющих к сети одинаковые требования, может прокладываться несколько маршрутов, чтобы за счет распараллеливания ускорить передачу данных.

Признаки потока могут иметь *глобальное* или *локальное* значение — в первом случае они однозначно определяют поток в пределах всей сети, а во втором — в пределах одного транзитного узла. Пара адресов конечных узлов для идентификации потока — это пример глобального признака. Примером признака, локально определяющего поток в пределах устройства, может служить номер (идентификатор) интерфейса данного устройства, на который поступили данные. Например, возвращаясь к рис. 2.14, узел *1* может быть настроен так, чтобы передавать на интерфейс *B* все данные, поступившие с интерфейса *A*, а на интерфейс *C* — данные, поступившие с интерфейса *D*. Такое правило позволяет отделить поток данных узла *2* от потока данных узла *7* и направлять их для транзитной передачи через разные узлы сети, в данном случае поток узла *2* — через узел *5*, а поток узла *7* — через узел *8*.

Метка потока — это особый тип признака. Она представляет собой некоторое число, которое несут все данные потока. **Глобальная метка** назначается данным потока и не меняет своего значения на всем протяжении его пути следования от узла источника до узла назначения, таким образом, она уникально определяет поток в пределах сети. В некоторых технологиях используются **локальные метки** потока, динамически меняющие свое значение при передаче данных от одного узла к другому.

Таким образом, распознавание потоков во время коммутации происходит на основании признаков, в качестве которых, помимо обязательного адреса назначения данных, могут выступать и другие признаки, такие, например, как идентификаторы приложений.

Маршрутизация

Задача маршрутизации, в свою очередь, включает в себя две подзадачи:

- определение маршрута;
- оповещение сети о выбранном маршруте.

Определить маршрут означает выбрать последовательность транзитных узлов и их интерфейсов, через которые надо передавать данные, чтобы доставить их адресату. Определение маршрута — сложная задача, особенно когда конфигурация сети такова, что между парой взаимодействующих сетевых интерфейсов существует множество путей. Чаще всего выбор останавливают на одном *оптимальном*[1] по некоторому критерию маршруте. В качестве критериев оптимальности могут выступать, например, номинальная пропускная способность и загруженность каналов связи; задержки, вносимые каналами; количество промежуточных транзитных узлов; надежность каналов и транзитных узлов.

[1] На практике для снижения объема вычислений ограничиваются поиском не оптимального в математическом смысле, а рационального, то есть близкого к оптимальному, маршрута.

Но даже в том случае, когда между конечными узлами существует только *один* путь, при сложной топологии сети его нахождение может представлять собой нетривиальную задачу.

Маршрут может определяться эмпирически («вручную») администратором сети на основании различных часто не формализуемых соображений. Среди побудительных мотивов выбора пути могут быть: особые требования к сети со стороны различных типов приложений, решение передавать трафик через сеть определенного поставщика услуг, предположения о пиковых нагрузках на некоторые каналы сети, соображения безопасности.

Однако эмпирический подход к определению маршрутов мало пригоден для большой сети со сложной топологией. В этом случае используются автоматические методы определения маршрутов. Для этого конечные узлы и другие устройства сети оснащаются специальными программными средствами, которые организуют взаимный обмен служебными сообщениями, позволяющий каждому узлу составить свое «представление» о сети. Затем на основе собранных данных программными методами определяются рациональные маршруты.

При выборе маршрута часто ограничиваются только информацией о топологии сети. Этот подход иллюстрирует рис. 2.15. Для передачи трафика между конечными узлами *A* и *C* существует два альтернативных маршрута: *A-1-2-3-C* и *A-1-3-C*. Если мы учитываем только топологию, то выбор очевиден — маршрут *A-1-3-C*, который имеет меньше транзитных узлов.

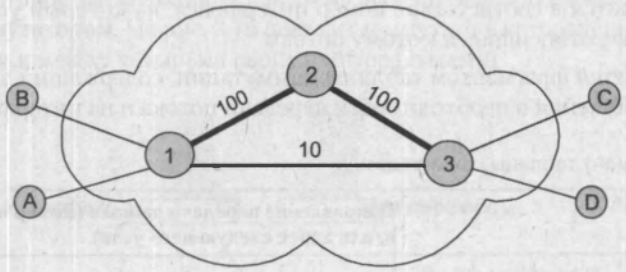


Рис. 2.15. Выбор маршрута

Решение было найдено путем минимизации критерия, в качестве которого в данном примере выступала длина маршрута, измеренная количеством транзитных узлов. Однако, возможно, наш выбор был не самым лучшим. На рисунке показано, что каналы *1-2* и *2-3* обладают пропускной способностью 100 Мбит/с, а канал *1-3* — только 10 Мбит/с. Если мы хотим, чтобы наша информация передавалась по сети с максимально возможной скоростью, то нам следовало бы выбрать маршрут *A-1-2-3-C*, хотя он и проходит через большее количество промежуточных узлов. То есть можно сказать, что маршрут *A-1-2-3-C* в данном случае оказывается «более коротким».

Абстрактный способ измерения степени близости между двумя объектами называется **метрикой**. Так, для измерения длины маршрута могут быть использованы разные метрики — количество транзитных узлов, как в предыдущем примере, линейная протяженность маршрута и даже его стоимость в денежном выражении. Для построения метрики, учитывающей пропускную способность, часто используют следующий прием: длину каждого канала-участка характеризуют величиной, обратной его пропускной способности. Чтобы оперировать целыми числами, выбирают некоторую константу, заведомо большую, чем

пропускные способности каналов в сети. Например, если мы в качестве такой константы выберем 100 Мбит/с, то метрика каждого из каналов *1-2* и *2-3* равна 1, а метрика канала *1-3* составляет 10. Метрика маршрута равна сумме метрик составляющих его каналов, поэтому часть пути *1-2-3* обладает метрикой 2, а альтернативная часть пути *1-3* — метрикой 10. Мы выбираем более «короткий» путь, то есть путь *A-1-2-3-C*.

Описанные подходы к выбору маршрутов не учитывают текущую степень загруженности каналов трафиком[1]. Используя аналогию с автомобильным трафиком, можно сказать, что мы выбирали маршрут по карте, учитывая количество промежуточных городов и ширину дороги (аналог пропускной способности канала), отдавая предпочтение скоростным магистралям. Но мы не стали слушать радио или телевизионную программу, которая сообщает о текущих заторах на дорогах. Так что наше решение оказывается отнюдь не лучшим, когда по маршруту *A-1-2-3-C* уже передается большое количество потоков, а маршрут *A-1-3-C* практически свободен.

После того как маршрут определен (вручную или автоматически), надо *оповестить* о нем все устройства сети. Сообщение о маршруте должно нести каждому транзитному устройству примерно такую информацию: «каждый раз, когда в устройство поступят данные, относящиеся к потоку *n*, их следует передать для дальнейшего продвижения на интерфейс *F*». Каждое подобное сообщение о маршруте обрабатывается устройством, в результате создается новая запись в **таблице коммутации**. В этой таблице локальному или глобальному признаку (признакам) потока (например, метке, номеру входного интерфейса или адресу назначения) ставится в соответствие номер интерфейса, на который устройство должно передавать данные, относящиеся к этому потоку.

Таблица 2.1 является фрагментом таблицы коммутации, содержащий запись, сделанную на основании сообщения о необходимости передачи потока *n* на интерфейс *F*.

Таблица 2.1. Фрагмент таблицы коммутации

Признаки потока	Направление передачи данных (номер интерфейса и/или адрес следующего узла)
...	...
n	F
...	...

Конечно, детальное описание структуры сообщения о маршруте и содержимого таблицы коммутации зависит от конкретной технологии, однако эти особенности не меняют сущности рассматриваемых процессов.

Передача информации транзитным устройствам о выбранных маршрутах, так же как и определение маршрута, может осуществляться вручную или автоматически. Администратор сети может зафиксировать маршрут, выполнив в ручном режиме конфигурирование устройства, например, жестко скоммутировав на длительное время определенные пары входных и выходных интерфейсов (как работали «телефонные барышни» на первых

[1] Такие методы, в которых используется информация о текущей загруженности каналов связи, позволяют определять более рациональные маршруты, однако требуют интенсивного обмена служебной информацией между узлами сети.

коммутаторах). Он может также по собственной инициативе внести запись о маршруте в таблицу коммутации.

Однако поскольку топология и состав информационных потоков могут меняться (отказы узлов или появление новых промежуточных узлов, изменение адресов или определение новых потоков), гибкое решение задач определения и задания маршрутов предполагает постоянный анализ состояния сети и обновление маршрутов и таблиц коммутации. В таких случаях задачи прокладки маршрутов, как правило, не могут быть решены без достаточно сложных программных и аппаратных средств.

Продвижение данных

Итак, пусть маршруты определены, записи о них сделаны в таблицах всех транзитных узлов, все готово к выполнению основной операции — передаче данных между абонентами (коммутации абонентов).

Для каждой пары абонентов эта операция может быть представлена несколькими (по числу транзитных узлов) *локальными* операциями коммутации. Прежде всего, отправитель должен выставить данные на тот свой интерфейс, с которого начинается найденный маршрут, а все транзитные узлы должны соответствующим образом выполнить «переброску» данных с одного своего интерфейса на другой, другими словами, выполнить **коммутацию интерфейсов**. Устройство, функциональным назначением которого является коммутация, называется **коммутатором**. На рис. 2.16 показан коммутатор, который переключает информационные потоки между четырьмя своими интерфейсами.

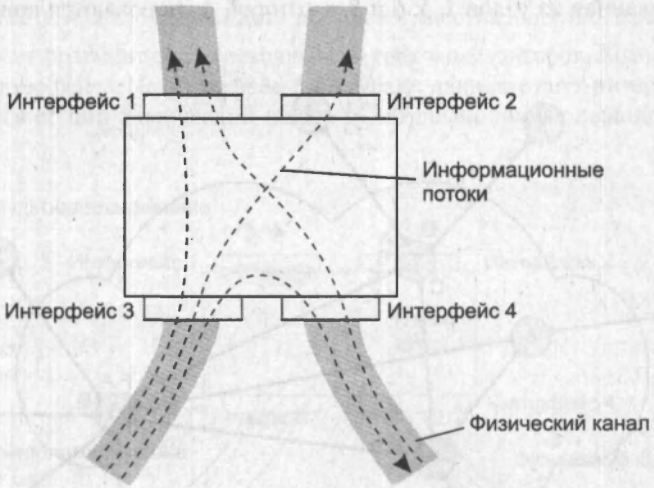


Рис. 2.16. Коммутатор

Однако прежде чем выполнить коммутацию, коммутатор должен распознать поток. Для этого поступившие данные анализируются на предмет наличия в них признаков какого-либо из потоков, заданных в таблице коммутации. Если произошло совпадение, то эти данные направляются на интерфейс, определенный для них в маршруте.

О ТЕРМИНАХ

Термины «коммутация», «таблица коммутации» и «коммутатор» в телекоммуникационных сетях могут трактоваться неоднозначно. Мы уже определили коммутацию как процесс соединения абонентов сети через транзитные узлы. Этим же термином мы обозначаем и соединение интерфейсов в пределах отдельного транзитного узла. Коммутатором в широком смысле называется устройство любого типа, способное выполнять операции переключения потока данных с одного интерфейса на другой. Операция коммутации может выполняться в соответствии с различными правилами и алгоритмами. Некоторые способы коммутации и соответствующие им таблицы и устройства получили специальные названия. Например, в технологиях сетевого уровня, таких как IP и IPX, для обозначения аналогичных понятий используются термины «маршрутизация», «таблица маршрутизации», «маршрутизатор». В то же время за другими специальными типами коммутации и соответствующими устройствами закрепились те же самые названия «коммутация», «таблица коммутации» и «коммутатор», применяемые в узком смысле, например, как коммутация и коммутатор локальной сети. Для телефонных сетей, которые появились намного раньше компьютерных, также характерна аналогичная терминология, коммутатор является здесь синонимом телефонной станции. Из-за солидного возраста и гораздо большей (пока) распространенности телефонных сетей чаще всего в телекоммуникациях под термином «коммутатор» понимают именно телефонный коммутатор.

Коммутатором может быть как специализированное устройство, так и универсальный компьютер со встроенным программным механизмом коммутации, в этом случае коммутатор называется программным. Компьютер может совмещать функции коммутации данных с выполнением своих обычных функций как конечного узла. Однако во многих случаях более рациональным является решение, в соответствии с которым некоторые узлы в сети выделяются *специально* для коммутации. Эти узлы образуют **коммутационную сеть**, к которой подключаются все остальные. На рис. 2.17 показана коммутационная сеть, образованная из узлов *1*, *5*, *6* и *8*, к которой подключаются конечные узлы *2*, *3*, *4*, *7*, *9* и *10*, *11*.

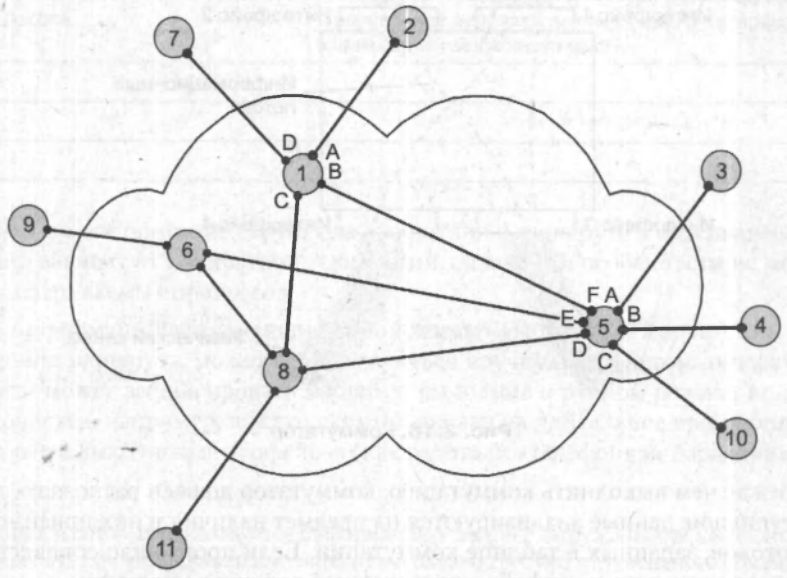


Рис. 2.17. Коммутационная сеть

Мультиплексирование и демультиплексирование

Чтобы определить, на какой интерфейс следует передать поступившие данные, коммутатор должен выяснить, к какому потоку они относятся. Эта задача должна решаться независимо от того, поступает на вход коммутатора только один «чистый» поток или «смешанный» поток, являющийся результатом агрегирования нескольких потоков. В последнем случае к задаче распознавания потоков добавляется задача демультиплексирования.

Демультиплексирование — разделение суммарного агрегированного потока на несколько составляющих его потоков.

Как правило, операцию коммутации сопровождает также обратная операция мультиплексирования.

Мультиплексирование — образование из нескольких отдельных потоков общего агрегированного потока, который передается по одному физическому каналу связи.

Другими словами, мультиплексирование — это способ разделения одного имеющегося физического канала между несколькими одновременно протекающими сеансами связи между абонентами сети.

Операции мультиплексирования/демультиплексирования имеют такое же важное значение в любой сети, как и операции коммутации, потому что без них пришлось бы для каждого потока предусматривать отдельный канал, что привело бы к большому количеству параллельных связей в сети и свело бы «на нет» все преимущества неполносвязной сети.

На рис. 2.18 показан фрагмент сети, состоящий из трех коммутаторов. Коммутатор *1* имеет четыре сетевых интерфейса. На интерфейс *1* поступают данные с двух интерфейсов — *3* и *4*. Их надо передать в общий физический канал, то есть выполнить операцию мультиплексирования.

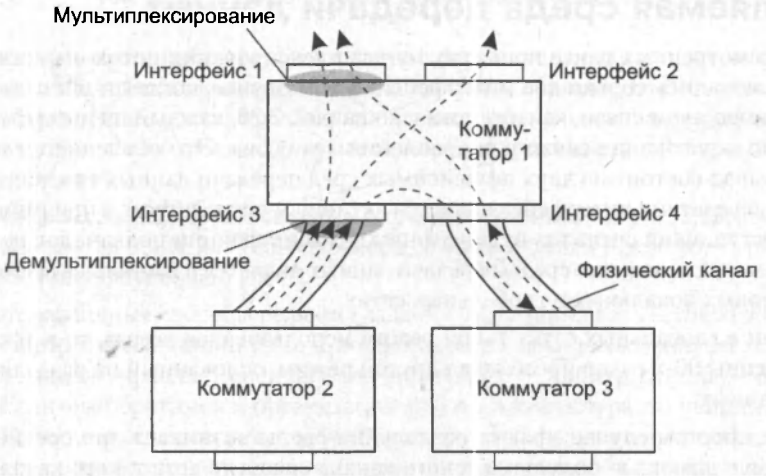


Рис. 2.18. Операции мультиплексирования и демультиплексирования потоков при коммутации

Одним из основных способов мультиплексирования потоков является **разделение времени**. При этом способе каждый поток время от времени (с фиксированным или случайным периодом) получает физический канал в полное свое распоряжение и передает по нему свои данные. Распространено также **частотное разделение** канала, когда каждый поток передает данные в выделенном ему частотном диапазоне.

Технология мультиплексирования должна позволять получателю такого суммарного потока выполнять обратную операцию — разделение (демультиплексирование) данных на слагаемые потоки. На интерфейсе *3* коммутатор выполняет демультиплексирование потока на три составляющих его подпотока. Один из них он передает на интерфейс *1*, другой — на интерфейс *2*, третий — на интерфейс *4*.

Вообще говоря, на каждом интерфейсе могут одновременно выполняться обе функции — мультиплексирование и демультиплексирование.

Частный случай коммутатора, у которого все входящие информационные потоки коммутируются на один выходной интерфейс, где они мультиплексируются в один агрегированный поток, называется **мультиплексором**. Коммутатор, который имеет один входной интерфейс и несколько выходных, называется **демультиплексором** (рис. 2.19).

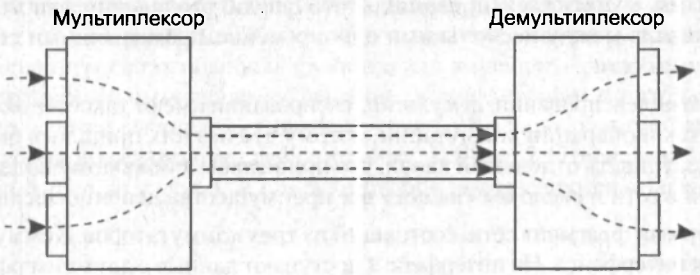


Рис. 2.19. Мультиплексор и демультиплексор

Разделяемая среда передачи данных

Во всех рассмотренных ранее примерах мультиплексирования потоков к каждой линии связи подключались только два интерфейса. В том случае, когда линия связи является дуплексным каналом связи, как это показано на рис. 2.20, каждый из интерфейсов монопольно использует канал связи в направлении «от себя». Это объясняется тем, что дуплексный канал состоит из двух независимых сред передачи данных (подканалов), и так как только передатчик интерфейса является активным устройством, а приемник пассивно ожидает поступления сигналов от приемника, то конкуренции подканалов не возникает. Такой режим использования среды передачи данных является в настоящее время основным в компьютерных локальных и глобальных сетях.

Однако если в глобальных сетях такой режим использовался всегда, то в локальных сетях до середины 90-х годов преобладал другой режим, основанный на разделяемой среде передачи данных.

В наиболее простом случае эффект разделения среды возникает при соединении двух интерфейсов с помощью полудуплексного канала связи, то есть такого канала, который может передавать данных в любом направлении, но только попеременно (рис. 2.21). В этом

случае к одной и той же среде передачи данных (например, к коаксиальному кабелю или общей радиосреде) подключены два приемника двух независимых узлов сети.

Разделяемой средой (shared medium) называется физическая среда передачи данных, к которой непосредственно подключено несколько передатчиков узлов сети. Причем в каждый момент времени только один из передатчиков какого-либо узла сети получает доступ к разделяемой среде и использует ее для передачи данных приемнику другого узла, подключенному к этой же среде.

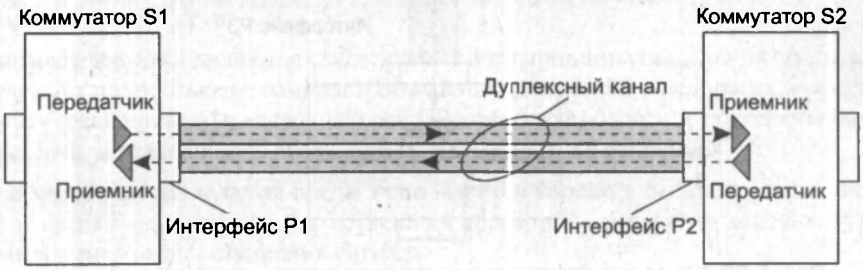


Рис. 2.20. Дуплексный канал — разделяемая среда отсутствует

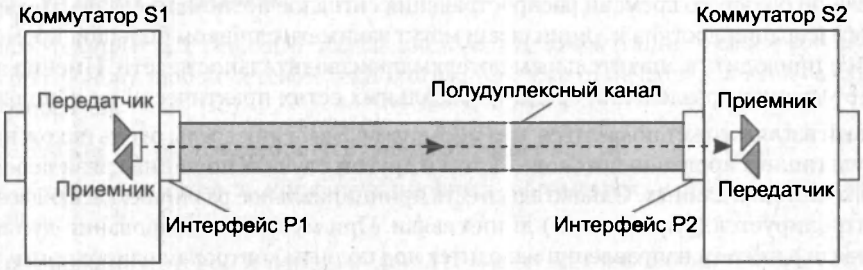


Рис. 2.21. Полудуплексный канал — разделяемая среда

При таком применении среды передачи данных возникает новая задача *совместного использования среды независимыми передатчиками* таким образом, чтобы в каждый отдельный момент времени по среде передавались данные только одного передатчика. Другими словами, возникает *необходимость в механизме синхронизации доступа интерфейсов к разделяемой среде.*

Обобщением разделяемой среды является случай, показанный на рис. 2.22, когда к каналу связи подключаются более двух интерфейсов (в приведенном примере — три), при этом применяется топология общей шины.

Существуют различные способы решения задачи организации совместного доступа к разделяемым линиям связи. Одни из них подразумевают *централизованный* подход, когда доступом к каналу управляет специальное устройство — **арбитр**, другие — *децентрализованный*. Если мы обратимся к организации работы компьютера, то увидим, что доступ к системной шине компьютера, которую совместно используют внутренние блоки компьютера, управляется централизованно — либо процессором, либо специальным арбитром шины.

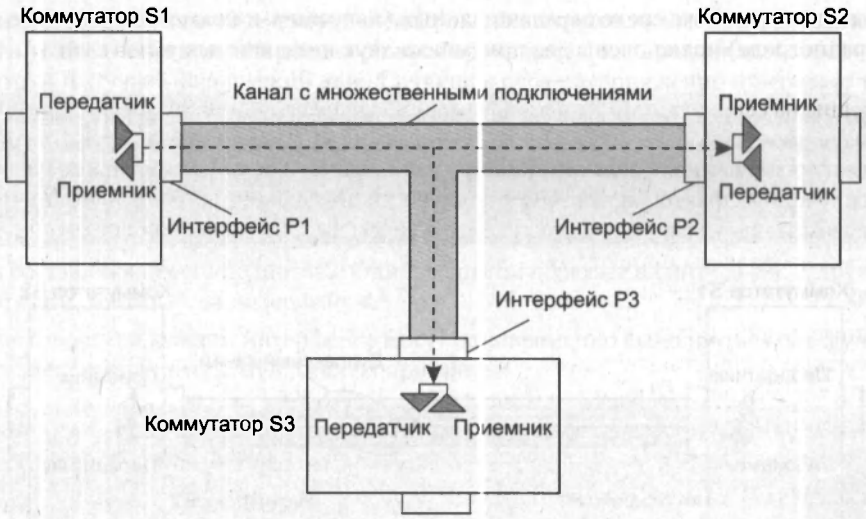


Рис. 2.22. Канал с множественными подключениями — разделяемая среда

В сетях организация совместного доступа к линиям связи имеет свою специфику из-за существенно большего времени распространения сигналов по линиям связи. Здесь процедуры согласования доступа к линии связи могут занимать слишком большой промежуток времени и приводить к значительным потерям производительности сети. Именно по этой причине механизм разделения среды в глобальных сетях практически не используется.

На первый взгляд может показаться, что механизм разделения среды очень похож на механизм мультиплексирования потоков — в том и другом случаях по линии связи передаются несколько потоков данных. Однако здесь есть принципиальное различие, касающееся того, как контролируется (управляется) линия связи. При мультиплексировании дуплексная линия связи в каждом направлении находится под полным контролем одного коммутатора, который решает, какие потоки разделяют общий канал связи.

Для локальных сетей разделяемая среда сравнительно долго была основным механизмом использования каналов связи, который применялся во всех технологиях локальных сетей — Ethernet, ArcNet, Token Ring, FDDI. При этом в технологиях локальных сетей применялись децентрализованные методы доступа к среде, не требующие наличия арбитра в сети. Популярность техники разделения среды в локальных сетях объяснялась простотой и экономичностью аппаратных решений. Например, для создания сети Ethernet на коаксиальном кабеле никакого другого сетевого оборудования кроме сетевых адаптеров компьютеров и самого кабеля не требуется. Наращивание количества компьютеров в локальной сети Ethernet на коаксиальном кабеле выполняется также достаточно просто — путем присоединения нового отрезка кабеля к существующему.

Сегодня в проводных локальных сетях метод разделения среды практически перестал применяться. Основной причиной отказа от разделяемой среды явилась ее низкая и плохо предсказуемая производительность, а также плохая масштабируемость[1]. Низкая про-

[1] Масштабируемостью называют свойство сети допускать наращивание количества узлов и протяженность линий связи в очень широких пределах без снижения производительности.

изводительность объясняется тем, что пропускная способность канала связи делится между всеми компьютерами сети. Например, если локальная сеть Ethernet состоит из 100 компьютеров, а для их связи используются коаксиальный кабель и сетевые адаптеры, работающие на скорости 10 Мбит/с, то в среднем на каждый компьютер приходится только 0,1 Мбит/с пропускной способности. Более точно оценить долю пропускной способности, приходящуюся на какой-либо компьютер сети, трудно, так как эта величина зависит от многих случайных факторов, например активности других компьютеров. Наверно, к этому моменту читателю уже понятна причина плохой масштабируемости подобной сети — чем больше мы добавляем компьютеров, тем меньшая доля пропускной способности достается каждому компьютеру сети.

Описанные недостатки являются следствием самого принципа разделения среды, поэтому преодолеть их полностью невозможно. Появление в начале 90-х недорогих коммутаторов локальных сетей привело к настоящей революции в этой области, и постепенно коммутаторы вытеснили разделяемую среду полностью.

Сегодня механизм разделения среды используется только в беспроводных локальных сетях, где среда — радиоэфир — естественным образом соединяет все конечные узлы, находящиеся в зоне распространения сигнала.

Типы коммутации

Комплекс технических решений обобщенной задачи коммутации в своей совокупности составляет основу любой сетевой технологии. Как уже отмечалось, к этим частным задачам относятся:

- определение потоков и соответствующих маршрутов;
- фиксация маршрутов в конфигурационных параметрах и таблицах сетевых устройств;
- распознавание потоков и передача данных между интерфейсами одного устройства;
- мультиплексирование/демультиплексирование потоков;
- разделение среды передачи.

Среди множества возможных подходов к решению задачи коммутации абонентов в сетях выделяют два основополагающих, к которым относят **коммутацию каналов** и **коммутацию пакетов**.

Каждый из этих двух подходов имеет свои достоинства и недостатки. Существуют традиционные области применения каждой из техник коммутации, например, телефонные сети строились и продолжают строиться с использованием техники коммутации каналов, а компьютерные сети в подавляющем большинстве основаны на технике коммутации пакетов. Техника коммутации пакетов гораздо моложе своей конкурентки и пытается вытеснить ее из некоторых областей, например из телефонии (в форме интернет- или IP-телефонии), но этот спор пока не решен, и, скорее всего, две техники коммутации будут сосуществовать еще долгое время, дополняя друг друга. Тем не менее по долгосрочным прогнозам многих специалистов будущее принадлежит технике коммутации пакетов, как более гибкой и универсальной.

ПРИМЕР-АНАЛОГИЯ

Поясним достаточно абстрактное описание обобщенной модели коммутации на примере работы традиционной почтовой службы. Почта также работает с информационными потоками, которые в данном случае составляют почтовые отправления. Основным признаком почтового потока является адрес получателя. Для упрощения будем рассматривать в качестве адреса только страну, например Индия, Норвегия, Россия, Бразилия и т. д. Дополнительным признаком потока может служить особое требование к надежности или скорости доставки. Например, пометка «Avia» на почтовых отправлениях в Бразилию выделит из общего потока почты в Бразилию подпоток, который будет доставляться самолетом.

Для каждого потока почтовая служба должна определить маршрут, который будет проходить через последовательность почтовых отделений, являющихся аналогами коммутаторов. В результате многолетней работы почтовой службы уже определены маршруты для большинства адресов назначения. Иногда возникают новые маршруты, связанные с появлением новых возможностей — политических, транспортных, экономических. После выбора нового маршрута нужно оповестить о нем сеть почтовых отделений. Как видно, эти действия очень напоминают работу телекоммуникационной сети. Информация о выбранных маршрутах следования почты представлена в каждом почтовом отделении в виде таблицы, в которой задано соответствие между страной назначения и следующим почтовым отделением. Например, в почтовом отделении города Саратова все письма, адресованные в Индию, направляются в почтовое отделение Ашхабада, а письма, адресованные в Норвегию, — в почтовое отделение Санкт-Петербурга. Такая таблица направлений доставки почты является прямой аналогией таблицы коммутации коммуникационной сети.

Каждое почтовое отделение работает подобно коммутатору. Все поступающие от абонентов и других почтовых отделений почтовые отправления сортируются, то есть происходит распознавание потоков. После этого почтовые отправления, принадлежащие одному «потоку», упаковываются в мешок, для которого в соответствии с таблицей направлений определяется следующее по маршруту почтовое отделение.

Выводы

Для того чтобы пользователь сети получил возможность доступа к ресурсам «чужих» компьютеров, таких как диски, принтеры, плоттеры, необходимо дополнить все компьютеры сети специальными средствами. В каждом компьютере функции передачи данных в линию связи выполняют совместно аппаратный модуль, называемый сетевым адаптером или сетевой интерфейсной картой, и управляющая программа — драйвер. Задачи более высокого уровня — формирование запросов к ресурсам и их выполнение — решают соответственно клиентские и серверные модули ОС.

Даже в простейшей сети, состоящей из двух компьютеров, возникают проблемы физической передачи сигналов по линиям связи: кодирование и модуляция, синхронизация передающего и принимающего устройств, контроль корректности переданных данных.

Важными характеристиками, связанными с передачей трафика через физические каналы, являются: предложенная нагрузка, скорость передачи данных, пропускная способность, емкость канала связи, полоса пропускания.

При связывании в сеть более двух компьютеров возникают проблемы выбора топологии (полносвязной, звезды, кольца, общей шины, иерархического дерева, произвольной); способа адресации (плоского или иерархического, числового или символьного); способа разделения линий связи и механизма коммутации.

В неполносвязных сетях соединение пользователей осуществляется путем коммутации через сеть транзитных узлов. При этом должны быть решены следующие задачи: определение потоков данных и маршрутов для них, продвижение данных в каждом транзитном узле, мультиплексирование и демультиплексирование потоков.

Среди множества возможных подходов к решению задачи коммутации выделяют два основополагающих — коммутацию каналов и пакетов.

Вопросы и задания

1. С какими ресурсами компьютера могут совместно работать несколько пользователей сети? Приведите примеры, когда у пользователей возникает необходимость разделять процессор?
2. Какие из перечисленных понятий могут быть определены как «набор информационных сообщений определенного формата, которыми обмениваются два устройства или две программы, а также алгоритм обмена этими сообщениями»:

   а) порт; б) протокол; в) логический интерфейс; г) физический интерфейс.
3. Опишите роль буферизации данных в процедуре доступа приложения, выполняемого на одном компьютере сети, к периферийному устройству другого компьютера. Сколько раз данные буферизуются при этом? Какой размер должен иметь буфер в каждом из таких случаев?
4. Что из перечисленного можно считать одним из возможных определений понятия «веб-сервер»:

   а) распределенная программа;

   б) часть веб-службы;

   в) клиентская часть распределенного сетевого приложения;

   г) браузер;

   д) локальное приложение;

   е) клиентская часть централизованного сетевого приложения;

   ж) серверная часть распределенного сетевого приложения;

   з) компьютер.
5. Приведите примеры сетевых служб. Какие из них ориентированы на администратора сети? Какие из них обычно входят в состав сетевой ОС?
6. Какие из перечисленных терминов в некотором контексте могут использоваться как синонимы:

   а) емкость канала связи;

   б) скорость передачи данных;

   в) полоса пропускания канала связи;

   г) пропускная способность канала связи.
7. Какие соображения следует учитывать при выборе топологии сети? Приведите достоинства и недостатки каждой из типовых топологий.
8. К какому типу относится каждый из восьми вариантов топологии на рис. 2.9. Для определенности рассматривайте приведенные варианты топологии построчно сверху вниз, слева направо.
9. Каким типом адреса снабжают посылаемые данные, когда хотят, чтобы они были доставлены всем узлам сети:

   а) multicast; б) anycast; в) broadcast; г) unicast.
10. В соответствии с классификацией адресов, используемых в компьютерных сетях, существуют символьные, числовые адреса, плоские, иерархические, индивидуальные, групповые и широковещательные адреса, а также адреса групповой рассылки. Как

бы вы классифицировали в приведенных терминах обычный почтовый адрес? Какой тип сетевого протокола соответствует процедуре определения адреса по почтовому индексу?

11. В чем состоит и как решается задача маршрутизации?
12. Работа почтового отделения во многом аналогична работе коммутатора компьютерной сети. Какие процедуры обработки почтовых отправлений соответствуют мультиплексированию? Демультиплексированию? Как создается и какую информацию содержит «таблица маршрутизации» почтового отделения? Какой атрибут информационного потока может служить аналогом пометки «АВИА» на почтовом конверте?
13. Опишите два основных подхода к организации совместного использования передающей среды несколькими передатчиками.
14. Приведите аргументы за и против использования разделяемой среды в LAN и WAN.

ГЛАВА 3 Коммутация каналов и пакетов

В этой главе продолжается исследование общих принципов коммутации в телекоммуникационных сетях. Мы детально изучим и сравним два основных типа коммутации — коммутацию каналов и коммутацию пакетов.

Исторически коммутация каналов появилась намного раньше коммутации пакетов и ведет свое происхождение от первых телефонных сетей. Невозможность динамического перераспределения пропускной способности физического канала является принципиальным ограничением сети с коммутацией каналов.

Принцип коммутации пакетов был изобретен разработчиками компьютерных сетей. При коммутации пакетов учитываются особенности компьютерного трафика, поэтому данный способ коммутации является более эффективным для компьютерных сетей по сравнению с традиционным методом коммутации каналов, применяющимся в телефонных сетях.

Однако достоинства и недостатки любой сетевой технологии — относительны. Наличие буферной памяти в коммутаторах пакетных сетей позволяет эффективно использовать пропускную способность каналов при передаче пульсирующего трафика, но приводит к случайным задержкам в доставке пакетов, что является недостатком для трафика реального времени, который традиционно передается с помощью техники коммутации каналов.

В этой главе рассматриваются три метода продвижения пакетов, используемые в сетях с коммутацией пакетов: дейтаграммная передача, передача с установлением логического соединения и техника виртуальных каналов.

В заключение главы рассматривается пример сети, построенной на стандартной технологии коммутации пакетов Ethernet.

Коммутация каналов

Сети, построенные на принципе коммутации каналов, имеют богатую историю, они и сегодня нашли широкое применение в мире телекоммуникаций, являясь основой создания высокоскоростных магистральных каналов связи. Первые сеансы связи между компьютерами были осуществлены через телефонную сеть, то есть также с применением техники коммутации каналов, а пользователи, которые получают доступ в Интернет по модему, продолжают обслуживаться этими сетями, так как их данные доходят до оборудования провайдера по местной телефонной сети.

В сетях с коммутацией каналов решаются все те частные задачи коммутации, которые были сформулированы ранее. Так, в качестве информационных потоков в сетях с коммутацией каналов выступают данные, которыми обмениваются пары **абонентов**[1]. Соответственно глобальным признаком потока является пара адресов (телефонных номеров) абонентов, связывающихся между собой. Для всех возможных потоков заранее определяются маршруты. Маршруты в сетях с коммутацией каналов задаются либо «вручную» администратором сети, либо находятся автоматически с привлечением специальных программных и аппаратных средств. Маршруты фиксируются в таблицах, в которых признакам потока ставятся в соответствие идентификаторы выходных интерфейсов коммутаторов. На основании этих таблиц происходит продвижение и мультиплексирование данных. Однако, как уже было сказано, в сетях с коммутацией каналов решение всех этих задач имеет свои особенности.

Элементарный канал

Одной из особенностей сетей с коммутацией каналов является понятие элементарного канала.

Элементарный канал (или просто **канал**) — это базовая техническая характеристика сети с коммутацией каналов, представляющая собой некоторое фиксированное в пределах данного типа сетей значение пропускной способности. Любая линия связи в сети с коммутацией каналов имеет пропускную способность, кратную элементарному каналу, принятому для данного типа сети.

В предыдущих разделах мы использовали термин «канал» как синоним термина «линия связи». Говоря же о сетях с коммутацией каналов, мы придаем термину «канал» значение единицы пропускной способности.

Значение элементарного канала, или, другими словами, минимальная единица пропускной способности линии связи, выбирается с учетом разных факторов. Очевидно, однако, что элементарный канал не стоит выбирать меньше минимально необходимой пропускной способности для передачи ожидаемой предложенной нагрузки. Например, в традиционных телефонных сетях наиболее распространенным значением элементарного канала сегодня является скорость 64 Кбит/с — это минимально достаточная скорость для качественной цифровой передачи голоса.

---

[1] Термин «абонент» принят в телефонии для обозначения конечного узла. Так как все мы — многолетние пользователи телефонной сети, то далее мы будем сопровождать наше объяснение принципа работы сетей с коммутацией каналов примерами из области телефонии.

ОЦИФРОВЫВАНИЕ ГОЛОСА

Задача оцифровывания голоса является частным случаем более общей проблемы — передачи аналоговой информации в дискретной форме. Она была решена в 60-е годы, когда голос начал передаваться по телефонным сетям в виде последовательности единиц и нулей. Такое преобразование основано на дискретизации непрерывных процессов как по амплитуде, так и по времени (рис. 3.1).

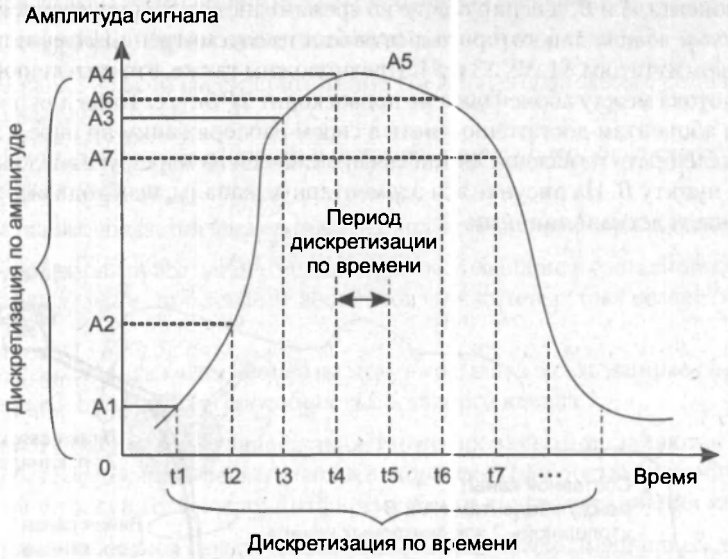


Рис. 3.1. Дискретная модуляция непрерывного процесса

Амплитуда исходной непрерывной функции измеряется с заданным периодом — за счет этого происходит *дискретизация по времени*. Затем каждый замер представляется в виде двоичного числа определенной разрядности, что означает *дискретизацию по значениям* — непрерывное множество возможных значений амплитуды заменяется дискретным множеством ее значений.

Для качественной передачи голоса используется частота квантования амплитуды звуковых колебаний в 8000 Гц (дискретизация по времени с интервалом 125 мкс). Для представления амплитуды одного замера чаще всего используется 8 бит кода, что дает 256 градаций звукового сигнала (дискретизация по значениям). В этом случае для передачи одного голосового канала необходима пропускная способность 64 Кбит/с: 8000 × 8 = 64 000 бит/с или 64 Кбит/с. Такой голосовой канал называют **элементарным каналом цифровых телефонных сетей**.

Линии связи в сетях с коммутацией пакетов (как, впрочем, и в остальных типах компьютерных сетей) имеют разную пропускную способность, одни — большую, другие — меньшую. Выбирая линии связи с разными скоростными качествами, специалисты, проектирующие сеть, стараются учесть разную интенсивность информационных потоков, которые могут возникнуть в разных фрагментах сети — чем ближе к центру сети, тем выше пропускная способность линии связи, так как магистральные линии агрегируют трафик большого количества периферийных линий связи.

Особенностью сетей с коммутацией каналов является то, что пропускная способность каждой линии связи должна быть равна *целому числу* элементарных каналов.

Так, линии связи, подключающие абонентов к телефонной сети, могут содержать 2, 24 или 30 элементарных каналов, а линии, соединяющие коммутаторы, — 480 или 1920 каналов.

Обратимся к фрагменту сети, изображенному на рис. 3.2. Предположим, что эта сеть характеризуется элементарным каналом P бит/с. В сети существуют линии связи разной пропускной способности, состоящие из 2, 3, 4 и 5 элементарных каналов. На рисунке показаны два абонента, A и B, генерирующие во время сеанса связи (телефонного разговора) *информационный поток*, для которого в сети был предусмотрен *маршрут*, проходящий через четыре коммутатора $S1$, $S2$, $S3$ и $S4$. Предположим также, что интенсивность информационного потока между абонентами не превосходит $2P$ бит/с. Тогда для обмена данными этим двум абонентам достаточно иметь в своем распоряжении по паре элементарных каналов, «выделенных» из каждой линии связи, лежащей на маршруте следования данных от пункта A к пункту B. На рисунке эти элементарные каналы, необходимые абонентам A и B, обозначены толстыми линиями.

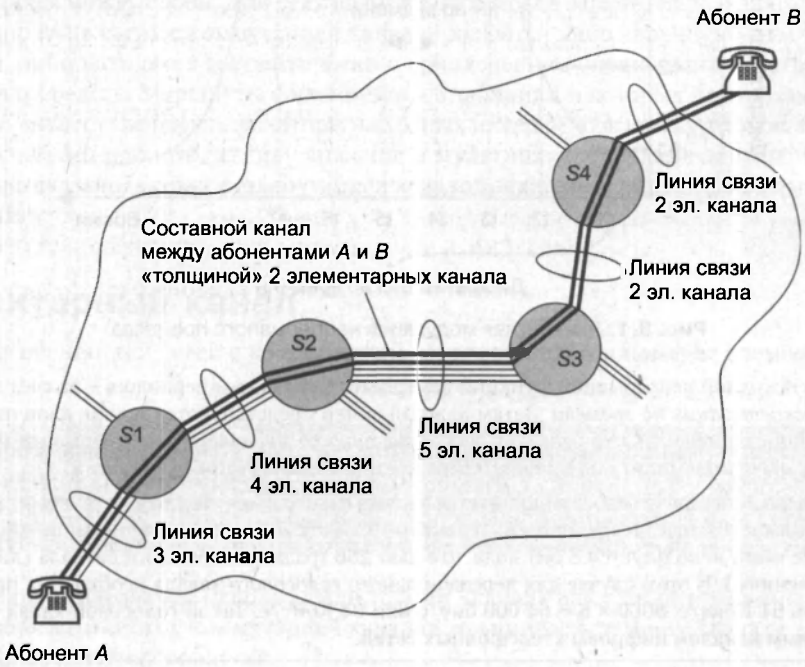


Рис. 3.2. Составной канал в сети с коммутацией каналов

Составной канал

Связь, построенную путем коммутации (соединения) элементарных каналов, называют **составным каналом**.

В рассматриваемом примере для соединения абонентов A и B был создан составной канал «толщиной» в два элементарных канала. Если изменить наше предположение и считать,

что предложенная нагрузка гарантированно не превысит P бит/с, то абонентам будет достаточно иметь в своем распоряжении составной канал, «толщиной» в один элементарный канал. В то же время абоненты, интенсивно обменивающиеся данными, могут предъявить и более высокие требования к пропускной способности составного канала. Для этого они должны в каждой линии связи зарезервировать за собой большее (но непременно одинаковое для всех линий связи) количество элементарных каналов.

Подчеркнем следующие свойства составного канала

- составной канал на всем своем протяжении состоит из *одинакового* количества элементарных каналов;
- составной канал имеет *постоянную и фиксированную пропускную способность* на всем своем протяжении;
- составной канал создается *временно* на период сеанса связи двух абонентов;
- на время сеанса связи все элементарные каналы, входящие в составной канал, поступают в *исключительное* пользование абонентов, для которых был создан этот составной канал;
- в течение всего сеанса связи абоненты могут посылать в сеть данные со скоростью, не превышающей пропускную способность составного канала;
- данные, поступившие в составной канал, гарантированно доставляются вызываемому абоненту *без задержек, потерь и с той же скоростью* (скоростью источника) вне зависимости от того, существуют ли в это время в сети другие соединения или нет;
- после окончания сеанса связи элементарные каналы, входившие в соответствующий составной канал, *объявляются свободными* и возвращаются в пул распределяемых ресурсов для использования другими абонентами.

В сети может одновременно происходить несколько сеансов связи (обычная ситуация для телефонной сети, в которой одновременно передаются разговоры сотен и тысяч абонентов). Разделение сети между сеансами связи происходит на уровне элементарных каналов. Например (см. рис. 3.2), мы можем предположить, что после того как в линии связи $S2$-$S3$ было выделено два канала для связи абонентов A и B, оставшиеся три элементарных канала были распределены между тремя другими сеансами связи, проходившими в это же время и через эту же линию связи. Такое *мультиплексирование* позволяет одновременно передавать через каждый физический канал трафик нескольких логических соединений.

Мультиплексирование означает, что абоненты вынуждены конкурировать за ресурсы, в данном случае за элементарные каналы. Возможны ситуации, когда некоторая промежуточная линия связи уже исчерпала свободные элементарные каналы, тогда новый сеанс связи, маршрут которого пролегает через данную линию связи, не может состояться.

Для того чтобы распознать такие ситуации, обмен данными в сети с коммутацией каналов предваряется **процедурой установления соединения**. В соответствии с этой процедурой абонент, являющийся инициатором сеанса связи (например, абонент A в нашей сети), посылает в коммутационную сеть **запрос**, представляющий собой сообщение, в котором содержится адрес вызываемого абонента, например абонента B[1].

[1] В телефонной сети посылке запроса соответствует набор телефонного номера.

Цель запроса — проверить, можно ли образовать составной канал между вызывающим и вызываемым абонентами. А для этого требуется соблюдение двух условий: наличие требуемого числа свободных элементарных каналов в каждой линии связи, лежащей на пути от *A* к *B*, и незанятость вызываемого абонента в другом соединении.

Запрос перемещается по *маршруту*, определенному для информационного потока данной пары абонентов. При этом используются глобальные таблицы коммутации, ставящие в соответствие *глобальному* признаку потока (адресу вызываемого абонента) идентификатор выходного интерфейса коммутатора (как уже упоминалось, такие таблицы часто называют также таблицами маршрутизации).

Если в результате прохождения запроса от абонента *A* к абоненту *B* выяснилось, что ничто не препятствует установлению соединения, происходит *фиксация* составного канала. Для этого во всех коммутаторах вдоль пути от *A* до *B* создаются записи в *локальных таблицах коммутации*, в которых указывается соответствие между *локальными признаками потока* — номерами элементарных каналов, зарезервированных для этого сеанса связи. Только после этого составной канал считается установленным, и абоненты *A* и *B* могут начать свой сеанс связи.

Таким образом, продвижение данных в сетях с коммутацией каналов происходит в два этапа:

1. В сеть поступает служебное сообщение — запрос, который несет адрес вызываемого абонента и организует создание составного канала.
2. По подготовленному составному каналу передается основной поток данных, для передачи которого уже не требуется никакой вспомогательной информации, в том числе адреса вызываемого абонента. Коммутация данных в коммутаторах выполняется на основе локальных признаков — номеров элементарных каналов.

Запросы на установление соединения не всегда завершаются успешно. Если на пути между вызывающим и вызываемым абонентами отсутствуют свободные элементарные каналы или вызываемый узел занят, то происходит **отказ в установлении соединения**. Например, если во время сеанса связи абонентов *A* и *B* абонент *C* пошлет запрос в сеть на установление соединения с абонентом *D*, то он получит отказ, потому что оба необходимых ему элементарных канала, составляющих линию связи коммутаторов *S*3 и *S*4, уже выделены соединению абонентов *A* и *B* (рис. 3.3). При отказе в установлении соединения сеть информирует вызывающего абонента специальным сообщением[1]. Чем больше нагрузка на сеть, то есть чем больше соединений она в данный момент поддерживает, тем больше вероятность отказа в удовлетворении запроса на установление нового соединения.

Мы описали процедуру установления соединения в *автоматическом динамическом режиме*, основанном на способности абонентов отправлять в сеть служебные сообщения — запросы на установление соединения и способности узлов сети обрабатывать такие сообщения. Подобный режим используется телефонными сетями: телефонный аппарат генерирует запрос, посылая в сеть импульсы (или тоновые сигналы), кодирующие номер вызываемого абонента, а сеть либо устанавливает соединение, либо сообщает об отказе сигналами «занято».

---

[1] Телефонная сеть в этом случае передает короткие гудки – сигнал «занято». Некоторые телефонные сети различают события «сеть занята» и «абонент занят», передавая гудки с разной частотой или используя разные тона.

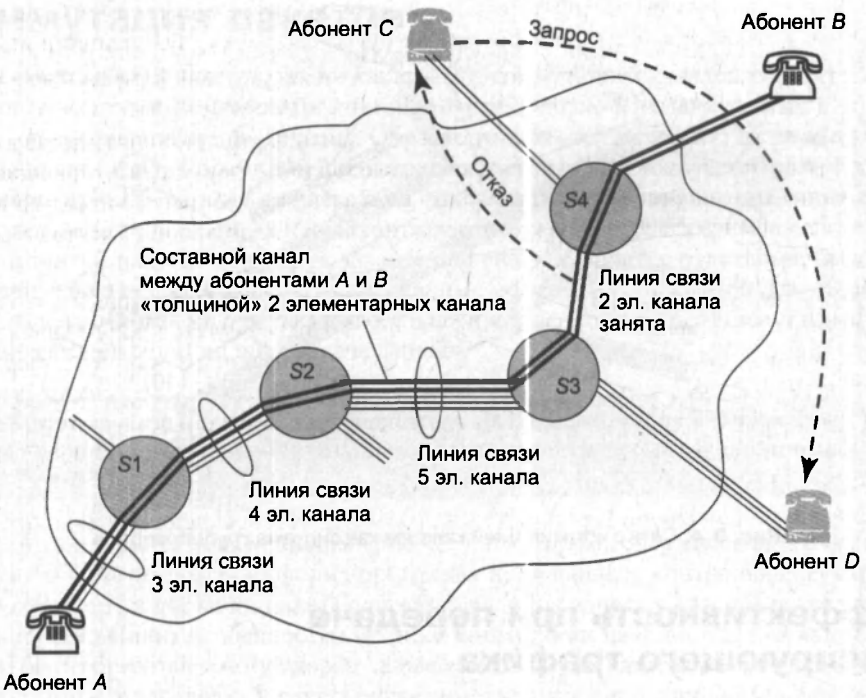


Рис. 3.3. Отказ в установлении соединения в сети с коммутацией каналов

Однако это — не единственно возможный режим работы сети с коммутацией каналов, существует и другой *статический ручной режим* установления соединения. Этот режим характерен для случаев, когда необходимо установить составной канал не на время одного сеанса связи абонентов, а на более долгий срок. Создание такого долговременного канала не могут инициировать абоненты, он создается администратором сети. Очевидно, что статический ручной режим мало пригоден для традиционной телефонной сети с ее короткими сеансами связи, однако он вполне оправдан для создания высокоскоростных телекоммуникационных каналов между городами и странами на более-менее постоянной основе.

Технология коммутации каналов ориентирована на минимизацию случайных событий в сети, то есть это технология, стремящаяся к детерминизму. Во избежание всяких возможных неопределенностей значительная часть работы по организации информационного обмена выполняется заранее, еще до того, как начнется собственно передача данных. Сначала по заданному адресу проверяется доступность необходимых элементарных каналов на всем пути от отправителя до адресата. Затем эти каналы закрепляются на все время сеанса для исключительного использования двумя абонентами и коммутируются в один непрерывный «трубопровод» (составной канал), имеющий «шлюзовые задвижки» на стороне каждого из абонентов. После этой исчерпывающей подготовительной работы остается сделать самое малое: «открыть шлюзы» и позволить информационному потоку свободно и без помех «перетекать» между заданными точками сети (рис. 3.4).

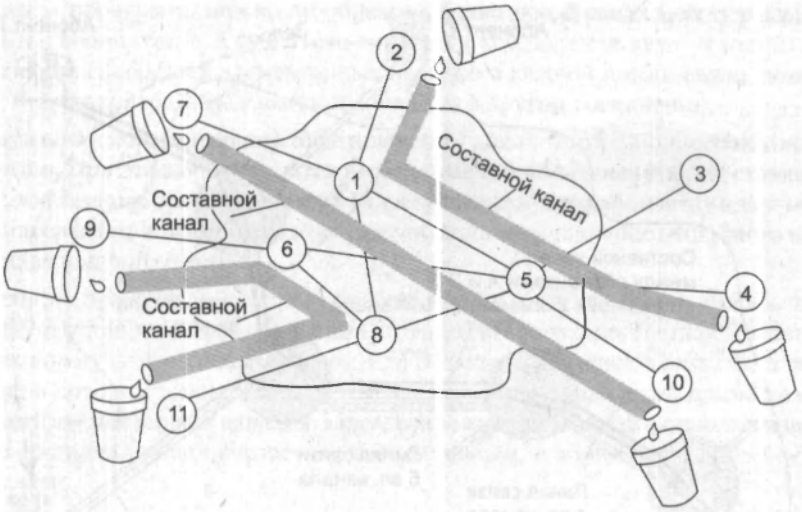


Рис. 3.4. Сеть с коммутацией каналов как система трубопроводов

Неэффективность при передаче пульсирующего трафика

Сети с коммутацией каналов наиболее эффективно передают пользовательский трафик в том случае, когда скорость его постоянна в течение всего сеанса связи и максимально соответствует *фиксированной* пропускной способности физических линий связи сети. Эффективность работы сети снижается, когда информационные потоки, генерируемые абонентами, приобретают *пульсирующий* характер.

Так, разговаривая по телефону, люди постоянно меняют темп речи, перемежая быстрые высказывания паузами. В результате соответствующие «голосовые» информационные потоки становятся неравномерными, а значит, снижается эффективность передачи данных. Правда, в случае телефонных разговоров это снижение оказывается вполне приемлемым и позволяет широко использовать сети с коммутацией каналов для передачи голосового трафика.

Гораздо сильнее снижает эффективность сети с коммутацией каналов передача так называемого *компьютерного трафика*, то есть трафика, генерируемого приложениями, с которыми работает пользователь компьютера. Этот трафик практически всегда является пульсирующим. Например, когда вы загружаете из Интернета очередную страницу, скорость трафика резко возрастает, а после окончания загрузки падает практически до нуля. Если для описанного сеанса доступа в Интернет вы задействуете сеть с коммутацией каналов, то большую часть времени составной канал между вашим компьютером и веб-сервером будет простаивать. В то же время часть производительности сети окажется закрепленной за вами и останется недоступной другим пользователям сети. Сеть в такие периоды похожа на пустой эскалатор метро, который движется, но полезную работу не выполняет, другими словами, «перевозит воздух».

Для эффективной передачи неравномерного компьютерного трафика была специально разработана техника коммутации пакетов.

Коммутация пакетов

Сети с коммутацией пакетов, так же как и сети с коммутацией каналов, состоят из коммутаторов, связанных физическими линиями связи. Однако передача данных в этих сетях происходит совершенно по-другому. Образно говоря, по сравнению с сетью с коммутацией каналов сеть с коммутацией пакетов ведет себя менее «ответственно». Например, она может принять данные для передачи, не заботясь о резервировании линий связи на пути следования этих данных и не гарантируя требуемую пропускную способность. Сеть с коммутацией пакетов не создает заранее для своих абонентов отдельных, выделенных исключительно для них каналов связи. Данные могут задерживаться и даже теряться по пути следования. Как же при таком хаосе и неопределенности сеть с коммутацией пакетов выполняет свои функции по передаче данных?

Важнейшим принципом функционирования сетей с коммутацией пакетов является представление информации, передаваемой по сети, в виде структурно отделенных друг от друга порций данных, называемых **пакетами**[1].

Каждый пакет снабжен **заголовком** (рис. 3.5), в котором содержится адрес назначения и другая вспомогательная информация (длина поля данных, контрольная сумма и др.), используемая для доставки пакета адресату. Наличие адреса в каждом пакете является одним из важнейших особенностей техники коммутации пакетов, так как каждый пакет может[2] быть обработан коммутатором *независимо* от других пакетов, составляющих сетевой трафик. Помимо заголовка у пакета может иметься еще одно дополнительное поле, размещаемое в конце пакета и поэтому называемое **концевиком**. В концевике обычно помещается **контрольная сумма**, которая позволяет проверить, была ли искажена информация при передаче через сеть или нет.

В зависимости от конкретной реализации технологии коммутации пакетов пакеты могут иметь фиксированную или переменную длину, кроме того, может меняться состав информации, размещенной в заголовках пакетов. Например, в технологии ATM пакеты (называемые там ячейками) имеют фиксированную длину, а в технологии Ethernet установлены лишь минимально и максимально возможные размеры пакетов (кадров).

Пакеты поступают в сеть *без предварительного резервирования линий связи* и *не с фиксированной заранее заданной скоростью*, как это делается в сетях с коммутацией каналов, а в том темпе, в котором их генерирует источник. Предполагается, что сеть с коммутацией пакетов, в отличие от сети с коммутацией каналов, всегда готова принять пакет от конечного узла.

Как и в сетях с коммутацией каналов, в сетях с коммутацией пакетов для каждого из потоков вручную или автоматически определяется маршрут, фиксируемый в хранящихся на коммутаторах таблицах коммутации. Пакеты, попадая на коммутатор, обрабатываются и направляются по тому или иному маршруту на основании информации, содержащейся в их заголовках, а также в таблице коммутации (рис. 3.6).

---

[1] Наряду с термином «пакет» используются также термины «кадр», «фрейм», «ячейка» и др. В данном контексте различия в значении этих терминов несущественны.

[2] В некоторых технологиях коммутации пакетов (например, в технологии виртуальных каналов) полная независимость обработки пакетов не обеспечивается.

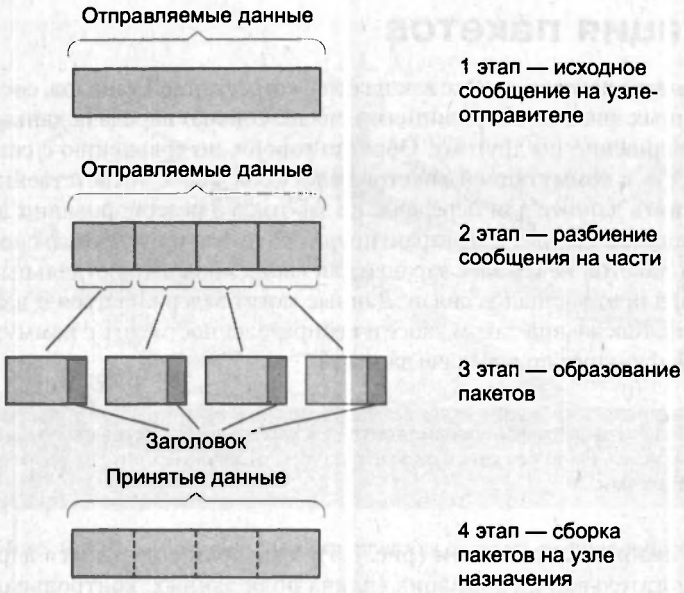


Рис. 3.5. Разбиение данных на пакеты

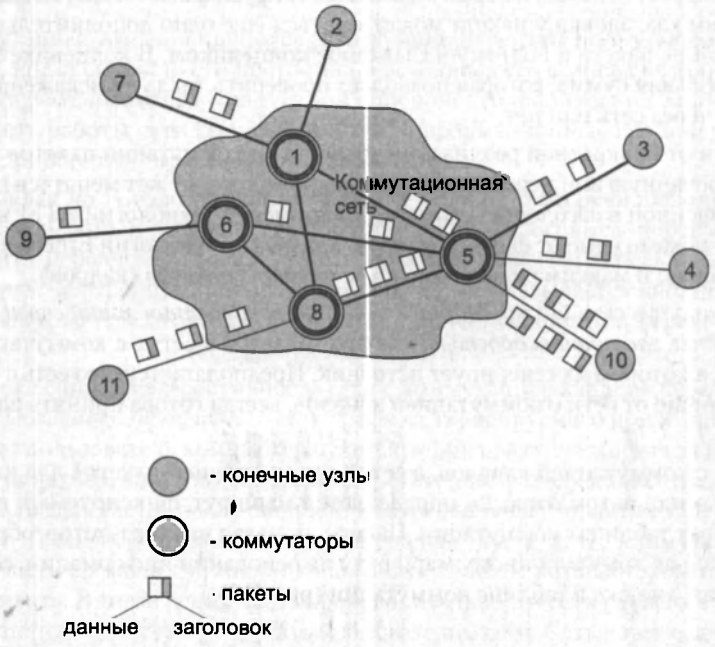


Рис. 3.6. Передача данных по сети в виде пакетов

ПРИМЕЧАНИЕ

Процедура резервирования пропускной способности может применяться и в пакетных сетях. Однако основная идея такого резервирования принципиально отличается от идеи резервирования пропускной способности в сетях с коммутацией каналов. Разница заключается в том, что пропускная способность канала сети с коммутацией пакетов может динамически перераспределяться между информационными потоками в зависимости от текущих потребностей каждого потока, чего не может обеспечить техника коммутации каналов. С деталями такого резервирования вы познакомитесь позже, в главе 7.

Пакеты, принадлежащие как одному и тому же, так и разным информационным потокам, при перемещении по сети могут «перемешиваться» между собой, образовывать очереди и «тормозить» друг друга. На пути пакетов могут встретиться линии связи, имеющие разную пропускную способность. В зависимости от времени суток может сильно меняться и степень загруженности линий связи. В таких условиях не исключены ситуации, когда пакеты, принадлежащими одному и тому же потоку, могут перемещаться по сети с разными скоростями и даже прийти к месту назначения не в том порядке, в котором они были отправлены.

Разделение данных на пакеты позволяет передавать неравномерный компьютерный трафик более эффективно, чем в сетях с коммутацией каналов. Это объясняется тем, что пульсации трафика от отдельных компьютеров носят случайный характер и распределяются во времени так, что их пики чаще всего не совпадают. Поэтому когда линия связи передает трафик большого количества конечных узлов, то в суммарном потоке пульсации сглаживаются, и пропускная способность линии используется более рационально, без длительных простоев. Это эффект иллюстрируется рис. 3.7, на котором показаны неравномерные потоки пакетов, поступающие от конечных узлов *3*, *4* и *10* в сети, изображенной на рис. 3.6.

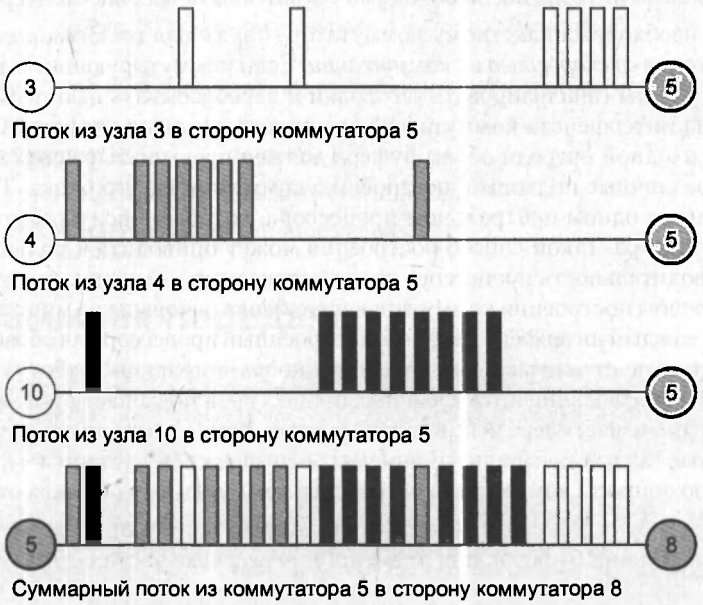


Рис. 3.7. Сглаживание трафика в сетях с коммутацией пакетов

Предположим, что эти потоки передаются в направлении коммутатора *8*, а следовательно, накладываются друг на друга при прохождении линии связи между коммутаторами *5* и *8*. Получающийся в результате суммарный поток является более равномерным, чем каждый из образующих его отдельных потоков.

Буферизация пакетов

Неопределенность и асинхронность перемещения данных в сетях с коммутацией пакетов предъявляет особые требования к работе коммутаторов в таких сетях.

Главное отличие пакетных коммутаторов[1] от коммутаторов в сетях с коммутацией каналов состоит в том, что они имеют внутреннюю **буферную память** для временного хранения пакетов.

Действительно, пакетный коммутатор не может принять решения о продвижении пакета, не имея в своей памяти всего пакета. Коммутатор проверяет контрольную сумму, и только если она говорит о том, что данные пакета не искажены, начинает обрабатывать пакет и по адресу назначения определяет следующий коммутатор. Поэтому *каждый* пакет последовательно бит за битом помещается во **входной буфер**. Имея в виду это свойство, говорят, что сети с коммутацией пакетов используют технику **сохранения с продвижением** (store-and-forward). Заметим, что для этой цели достаточно иметь буфер размером **в** один пакет.

Коммутатору нужны буферы *для согласования скоростей передачи данных в линиях связи*, подключенных к его интерфейсам. Действительно, если скорость поступления пакетов из одной линии связи в течение некоторого периода превышает пропускную способность той линии связи, в которую эти пакеты должны быть направлены, то во избежание потерь пакетов на целевом интерфейсе необходимо организовать выходную очередь (рис. 3.8).

Буферизация необходима пакетному коммутатору также *для согласования скорости поступления пакетов со скоростью их коммутации*. Если коммутирующий блок не успевает обрабатывать пакеты (анализировать заголовки и перебрасывать пакеты на нужный интерфейс), то на интерфейсах коммутатора возникают **входные очереди**. Очевидно, что для хранения входной очереди объем буфера должен превышать размер одного пакета. Существуют различные подходы к построению коммутирующего блока. Традиционный способ основан на одном центральном процессоре, который обслуживает все входные очереди коммутатора. Такой способ построения может приводить к большим очередям, так как производительность процессора разделяется между несколькими очередями. Современные способы построения коммутирующего блока основаны на многопроцессорном подходе, когда каждый интерфейс имеет свой встроенный процессор для обработки пакетов. Кроме того, существует центральный процессор, координирующий работу интерфейсных процессоров. Использование интерфейсных процессоров повышает производительность коммутатора и уменьшает очереди во входных интерфейсах. Однако такие очереди все равно могут возникать, так как центральный процессор по-прежнему остается «узким местом». Более подробно вопросы внутреннего устройства коммутаторов обсуждаются в главе 13.

Поскольку объем буферов в коммутаторах ограничен, иногда происходит потеря пакетов из-за переполнения буферов при временной перегрузке части сети, когда совпадают

[1] Для простоты будем далее называть коммутаторы сетей с коммутацией пакетов «пакетными коммутаторами».

периоды пульсации нескольких информационных потоков. Для сетей с коммутацией пакетов потеря пакетов является обычным явлением, и для компенсации таких потерь в данной сетевой технологии предусмотрен ряд специальных механизмов, которые мы рассмотрим позже.

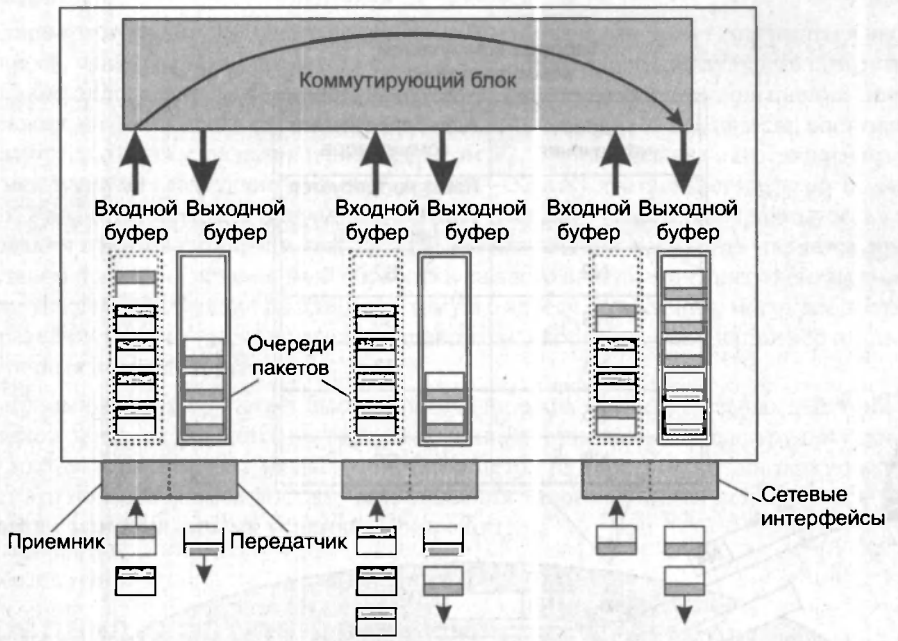


Рис. 3.8. Буферы и очереди пакетов в коммутаторе

Пакетный коммутатор может работать на основании одного из трех методов продвижения пакетов:

- дейтаграммная передача;
- передача с установлением логического соединения;
- передача с установлением виртуального канала.

Дейтаграммная передача

Дейтаграммный способ передачи данных основан на том, что все передаваемые пакеты *продвигаются* (передаются от одного узла сети другому) *независимо* друг от друга на основании одних и тех же правил.

Процедура обработки пакета определяется только значениями параметров, которые он несет в себе, и текущим состоянием сети (например, в зависимости от ее нагрузки пакет может стоять в очереди на обслуживание большее или меньшее время). Однако никакая информация об *уже переданных пакетах* сетью не хранится и в ходе обработки очередного пакета во внимание не принимается. То есть каждый отдельный пакет рассматривается сетью как совершенно независимая единица передачи — **дейтаграмма**.

Решение о продвижении пакета принимается на основе таблицы коммутации[1], ставящей в соответствие адресам назначения пакетов информацию, однозначно определяющую следующий по маршруту транзитный (или конечный) узел. В качестве такой информации могут выступать идентификаторы интерфейсов данного коммутатора или адреса входных интерфейсов коммутаторов, следующих по маршруту.

Таблица коммутации коммутатора S1

Адрес назначения	Адрес следующего коммутатора
N1	Пакет не требуется передавать через сеть
N2	S2
N3	S3
N4	S3
N5	S6
N6	S6

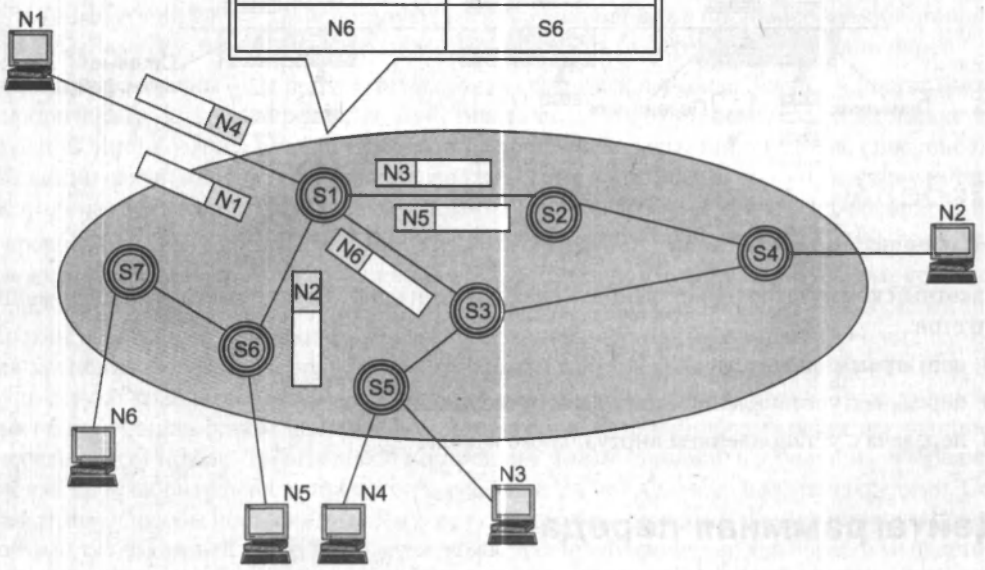


Рис. 3.9. Иллюстрация дейтаграммного принципа передачи пакетов

На рис. 3.9 показана сеть, в которой шесть конечных узлов (*N*1–*N*6) связаны семью коммутаторами (*S*1–*S*7). Показаны также несколько перемещающихся по разным маршрутам пакетов с разными адресами назначения (*N*1 - *N*6), на пути которых лежит коммутатор *S*1.

[1] Напомним, что в разных технологиях для обозначения таблиц, имеющих указанное выше функциональное назначение, могут использоваться другие термины (таблица маршрутизации, таблица продвижения и др.).

При поступлении каждого из этих пакетов в коммутатор *S*1 выполняется просмотр соответствующей таблицы коммутации и выбор дальнейшего пути перемещения. Так пакет с адресом *N*5 будет передан коммутатором *S*1 на интерфейс, ведущий к коммутатору *S*6, где в результате подобной процедуры этот пакет будут направлен конечному узлу получателю *N*5.

В таблице коммутации для одного и того же адреса назначения может содержаться несколько записей, указывающих соответственно на различные адреса следующего коммутатора. Такой подход называется **балансом нагрузки** и используется для повышения производительности и надежности сети. В примере, показанном на рис. 3.9, пакеты, поступающие в коммутатор *S*1 для узла назначения с адресом *N*2, в целях баланса нагрузки распределяются между двумя следующими коммутаторами — *S*2 и *S*3, что снижает нагрузку на каждый из них, а значит, сокращает очереди и ускоряет доставку. Некоторая «размытость» путей следования пакетов с одним и тем же адресом назначения через сеть является прямым следствием принципа независимой обработки каждого пакета, присущего дейтаграммному методу. Пакеты, следующие по одному и тому же адресу назначения, могут добираться до него разными путями также вследствие изменения состояния сети, например отказа промежуточных коммутаторов.

Дейтаграммный метод работает быстро, так как никаких предварительных действий перед отправкой данных проводить не требуется. Однако при таком методе трудно проверить факт доставки пакета узлу назначения. Этот метод не гарантирует доставку пакета, он делает это по мере возможности — для описания такого свойства используется термин **доставка с максимальными усилиями** (best effort).

Передача с установлением логического соединения

Следующий рассматриваемый нами способ продвижения пакетов основывается на знании устройствами сети «истории» обмена данными, например, на запоминании узлом-отправителем числа отправленных, а узлом-получателем — числа полученных пакетов. Такого рода информация фиксируется в рамках логического соединения.

Процедура согласования двумя конечными узлами сети некоторых параметров процесса обмена пакетами называется **установлением логического соединения**. Параметры, о которых договариваются два взаимодействующих узла, называются **параметрами логического соединения**.

Наличие логического соединения позволяет более рационально по сравнению с дейтаграммным способом обрабатывать пакеты. Например, при потере нескольких предыдущих пакетов может быть снижена скорость отправки последующих. Или благодаря нумерации пакетов и отслеживанию номеров отправленных и принятых пакетов можно повысить надежность путем отбрасывания дубликатов, упорядочивания поступивших и повторения передачи потерянных пакетов.

Параметры соединения могут быть: *постоянными*, то есть не изменяющимися в течение всего соединения (например, идентификатор соединения, способ шифрования пакета или максимальный размер поля данных пакета), или *переменными*, то есть динамически

отражающими текущее состояние соединения (например, последовательные номера передаваемых пакетов).

Когда отправитель и получатель *фиксируют* начало нового соединения, они, прежде всего, «договариваются» о начальных значениях параметров процедуры обмена и только после этого начинают передачу собственно данных.

Передача с установлением соединения более надежна, но требует больше времени для передачи данных и вычислительных затрат от конечных узлов, что иллюстрирует рис. 3.10.

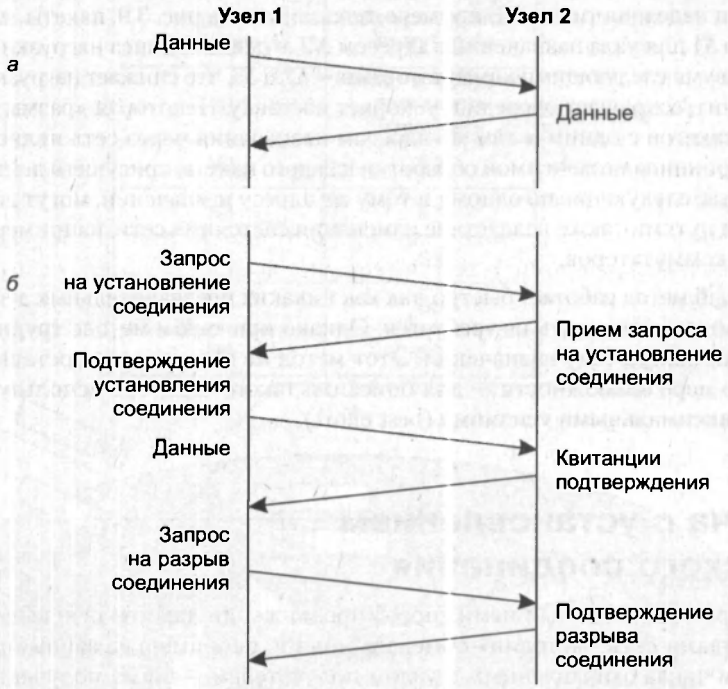


Рис. 3.10. Передача без установления соединения (*а*) и с установлением соединения (*б*)

Процедура установления соединения состоит обычно из трех шагов.

1. Узел-инициатор соединения отправляет узлу-получателю служебный пакет с предложением установить соединение.
2. Если узел-получатель согласен с этим, то он посылает в ответ другой служебный пакет, подтверждающий установление соединения и предлагающий некоторые параметры, которые будут использоваться в рамках данного логического соединения. Это могут быть, например, идентификатор соединения, количество кадров, которые можно отправить без получения подтверждения и т. п.
3. Узел-инициатор соединения может закончить процесс установления соединения отправкой третьего служебного пакета, в котором сообщит, что предложенные параметры ему подходят.

Логическое соединение может быть рассчитано на передачу данных как в одном направлении — от инициатора соединения, так и в обоих направлениях. После передачи некоторого

законченного набора данных, например определенного файла, узел-отправитель инициирует разрыв данного логического соединения, посылая соответствующий служебный кадр.

Заметим, что, в отличие от передачи дейтаграммного типа, в которой поддерживается только один тип кадра — информационный, передача с установлением соединения должна поддерживать как минимум два типа кадров — информационные кадры переносят собственно пользовательские данные, а служебные предназначаются для установления (разрыва) соединения.

После того как соединение установлено и все параметры согласованы, конечные узлы начинают передачу собственно данных. Пакеты данных обрабатываются коммутаторами точно так же, как и при дейтаграммной передаче: из заголовков пакетов извлекаются адреса назначения и сравниваются с записями в таблицах коммутации, содержащих информацию о следующих шагах по маршруту. Так же как дейтаграммы, пакеты, относящиеся к одному логическому соединению, в некоторых случаях (например, при отказе линии связи) могут доставляться адресату по разным маршрутам.

Однако передача с установлением соединения имеет важное отличие от дейтаграммной передачи, поскольку в ней помимо обработки пакетов на коммутаторах имеет место *дополнительная обработка пакетов на конечных узлах*. Например, если при установлении соединения была оговорена передача данных в зашифрованном виде, то шифрование пакетов выполняется узлом-отправителем, а расшифровка — узлом-получателем. Аналогично, для обеспечения в рамках логического соединения надежности всю работу по нумерации пакетов, отслеживанию номеров доставленных и недоставленных пакетов, посылки копий и отбрасывания дубликатов берут на себя конечные узлы.

ПРИМЕЧАНИЕ

Некоторые параметры логического соединения могут рассматриваться еще и как признаки информационного потока между узлами, установившими это логическое соединение.

Механизм установления логических соединений позволяет реализовывать дифференцированное обслуживание информационных потоков. Разное обслуживание могут получить даже потоки, относящиеся к одной и той же паре конечных узлов. Например, пара конечных узлов может установить два параллельно работающих логических соединения, в одном из которых передавать данные в зашифрованном виде, а в другом — открытым текстом.

Как видим, передача с установлением соединения предоставляет больше возможностей в плане надежности и безопасности обмена данными, чем дейтаграммная передача. Однако этот способ более медленный, так как он подразумевает дополнительные вычислительные затраты на установление и поддержание логического соединения.

Передача с установлением виртуального канала

Следующий способ продвижения данных основан на частном случае логического соединения, в число параметров которого входит жестко определенный для всех пакетов *маршрут*. То есть все пакеты, передаваемые в рамках данного соединения, должны проходить по одному и тому же закрепленному за этим соединением пути.

Единственный заранее проложенный фиксированный маршрут, соединяющий конечные узлы в сети с коммутацией пакетов, называют **виртуальным каналом** (virtual circuit или virtual channel).

Виртуальные каналы прокладываются для *устойчивых* информационных потоков. С целью выделения потока данных из общего трафика каждый пакет этого потока помечается специальным видом признака — **меткой**.

Так же как в сетях с установлением логических соединений, прокладка виртуального канала начинается с отправки из узла-источника специального пакета — запроса на установление соединения. В запросе указываются адрес назначения и метка потока, для которого прокладывается этот виртуальный канал. Запрос, проходя по сети, формирует новую запись в каждом из коммутаторов, расположенных на пути от отправителя до получателя. Запись говорит о том, каким образом коммутатор должен обслуживать пакет, имеющий заданную метку. Образованный виртуальный канал идентифицируется той же меткой[1].

После прокладки виртуального канала сеть может передавать по нему соответствующий поток данных. Во всех пакетах, которые переносят пользовательские данные, адрес назначения уже не указывается, его роль играет метка виртуального канала. При поступлении пакета на входной интерфейс коммутатор читает значение метки из заголовка пришедшего пакета и просматривает свою таблицу коммутации, по которой определяет, на какой выходной порт передать пришедший пакет.

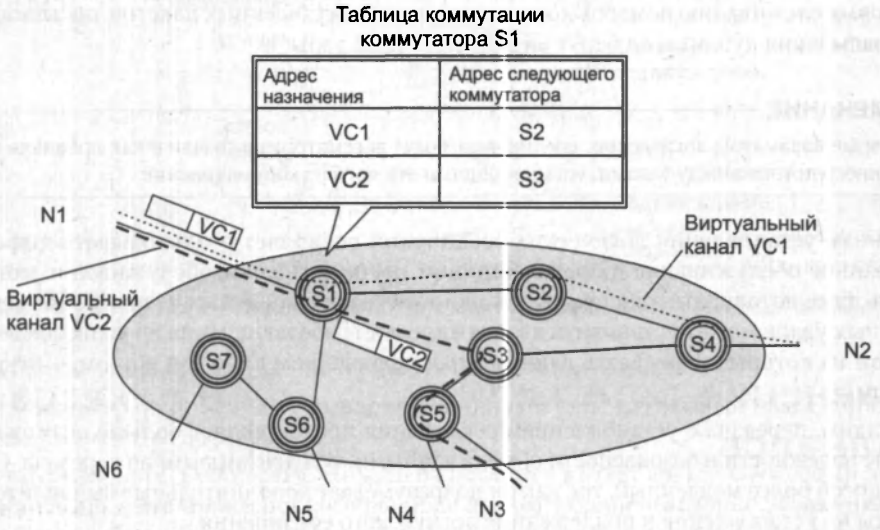


Рис. 3.11. Иллюстрация принципа работы виртуального канала

[1] Эта метка в различных технологиях называется по-разному: номер логического канала (Logical Channel Number, LCN) в технологии X.25, идентификатор соединения уровня канала данных (Data Link Connection Identifier, DLCI) в технологии Frame Relay, идентификатор виртуального канала (Virtual Channel Identifier, VCI) в технологии ATM.

На рис. 3.11 показана сеть, в которой проложены два виртуальных канала (Virtual Channel, VC), идентифицируемых метками VC1 и VC2. Первый проходит от конечного узла с адресом *N*1 до конечного узла с адресом *N*2 через промежуточные коммутаторы *S*1 и *S*2. Второй виртуальный канал VC2 обеспечивает продвижение данных по пути *N*1-*S*1-*S*3-*S*5-*N*3. В общем случае, между двумя конечными узлами может быть проложено несколько виртуальных каналов, например еще один виртуальный канал между узлами *N*1 и *N*2 мог бы проходить через промежуточный коммутатор *S*3. На рисунке показаны два пакета, несущие в своих заголовках метки потоков VC1 и VC2, которые играют роль адресов назначения.

Таблица коммутации в сетях, использующих виртуальные каналы, отличается от таблицы коммутации в дейтаграммных сетях. Она содержит записи *только о проходящих через коммутатор виртуальных каналах*, а не обо всех возможных адресах назначения, как это имеет место в сетях с дейтаграммным алгоритмом продвижения. Обычно в крупной сети количество проложенных через узел виртуальных каналов существенно меньше общего количества узлов, поэтому и таблицы коммутации в этом случае намного короче, а следовательно, анализ такой таблицы занимает у коммутатора меньше времени. По той же причине метка короче адреса конечного узла, и заголовок пакета в сетях с виртуальными каналами переносит по сети вместо длинного адреса компактный идентификатор потока.

ПРИМЕЧАНИЕ

Использование в сетях техники виртуальных каналов не делает их сетями с коммутацией каналов. Хотя в подобных сетях применяется процедура предварительного установления канала, этот канал является виртуальным, то есть по нему передаются отдельные пакеты, а не потоки информации с постоянной скоростью, как в сетях с коммутацией каналов.

В одной и той же сетевой технологии могут быть задействованы разные способы продвижения данных. Так, дейтаграммный протокол IP используется для передачи данных между отдельными сетями, составляющими Интернет. В то же время обеспечением надежной доставки данных между конечными узлами этой сети занимается протокол TCP, устанавливающий логические соединения без фиксации маршрута. И наконец, Интернет — это пример сети, применяющей технику виртуальных каналов, так как в состав Интернета входит немало сетей ATM и Frame Relay, поддерживающих виртуальные каналы.

Сравнение сетей с коммутацией пакетов и каналов

Прежде чем проводить техническое сравнение сетей с коммутацией пакетов и сетей с коммутацией каналов, проведем их неформальное сравнение на основе, как нам кажется, весьма продуктивной транспортной аналогии.

Транспортная аналогия для сетей с коммутацией пакетов и каналов

Для начала убедимся, что движение на дорогах имеет много общего с перемещением пакетов в сети *с коммутацией пакетов*.

Пусть автомобили в этой аналогии соответствуют пакетам, дороги — каналам связи, а перекрестки — коммутаторам. Подобно пакетам, автомобили перемещаются независимо друг от друга, разделяя пропускную способность дорог и создавая препятствия друг другу. Слишком интенсивный трафик, не соответствующий пропускной способности дороги, приводит к перегруженности дорог, в результате автомобили стоят в пробках, что соответствует очередям пакетов в коммутаторах.

На перекрестках происходит «коммутация» потоков автомобилей, каждый из автомобилей выбирает подходящее направление перекрестка, чтобы попасть в пункт назначения. Конечно, перекресток играет намного более пассивную роль по сравнению с коммутатором пакетов. Его активное участие в обработке трафика можно заметить только на регулируемых перекрестках, где светофор определяет очередность пересечения перекрестка потоками автомобилей. Еще активней, естественно, поведение регулировщика трафика, который может выбрать для продвижения не только поток автомобилей в целом, но и отдельный автомобиль.

Как и в сетях с коммутацией пакетов, к образованию заторов на дорогах приводит неравномерность движения автомобилей. Так, даже кратковременное снижение скорости одного автомобиля на узкой дороге может создать большую пробку, которой бы не было, если бы все автомобили всегда двигались с одной и той же скоростью и равными интервалами.

А теперь попробуем найти общее в автомобильном движении и в сетях с *коммутацией каналов*.

Иногда на дороге возникает ситуация, когда нужно обеспечить особые условия для движения колонны автомобилей. Например, представим, что очень длинная колонна автобусов перевозит детей из города в летний лагерь по многополосному шоссе. Для того чтобы колонна двигалась без препятствий, заранее для ее движения разрабатывается маршрут.

Затем на протяжении всего этого маршрута, который пересекает несколько перекрестков, для колонны выделяется отдельная полоса на всех отрезках шоссе. При этом полоса освобождается от другого трафика еще за некоторое время до начала движения колонны, и это резервирование отменяется только после того, как колонна достигает пункта назначения.

Во время движения все автомобили колонны едут с одинаковой скоростью и приблизительно равными интервалами между собой, не создавая препятствий друг другу. Очевидно, что для колонны автомобилей создаются наиболее благоприятные условия движения, но при этом автомобили теряют свою самостоятельность, превращаясь в поток, из которого нельзя «свернуть» в сторону. Дорога при такой организации движения используется нерационально, так как полоса простаивает значительную часть времени, как и полоса пропускания в сетях с коммутацией каналов.

Количественное сравнение задержек

Вернемся от автомобилей к сетевому трафику. Пусть пользователю сети необходимо передать достаточно неравномерный трафик, состоящий из периодов активности и пауз. Представим также, что он может выбрать, через какую сеть, с коммутацией каналов или пакетов, передавать свой трафик, причем в обеих сетях производительность каналов связи

одинаковы. Очевидно, что более эффективной с точки зрения временных затрат для нашего пользователя была бы работа в сети с коммутацией каналов, где ему в единоличное владение предоставляется зарезервированный канал связи. При этом способе все данные поступали бы адресату без задержки. Тот факт, что значительную часть времени зарезервированный канал будет простаивать (во время пауз), нашего пользователя не волнует — ему важно быстро решить собственную задачу.

Если бы пользователь обратился к услугам сети с коммутацией пакетов, то процесс передачи данных оказался бы более медленным, так как его пакеты, вероятно, не раз задерживались бы в очередях, ожидая освобождения необходимых сетевых ресурсов наравне с пакетами других абонентов.

Давайте рассмотрим более детально механизм возникновения задержек при передаче данных в сетях обоих типов. Пусть от конечного узла *N*1 отправляется сообщение к конечному узлу *N*2 (рис. 3.12). На пути передачи данных расположены два коммутатора.

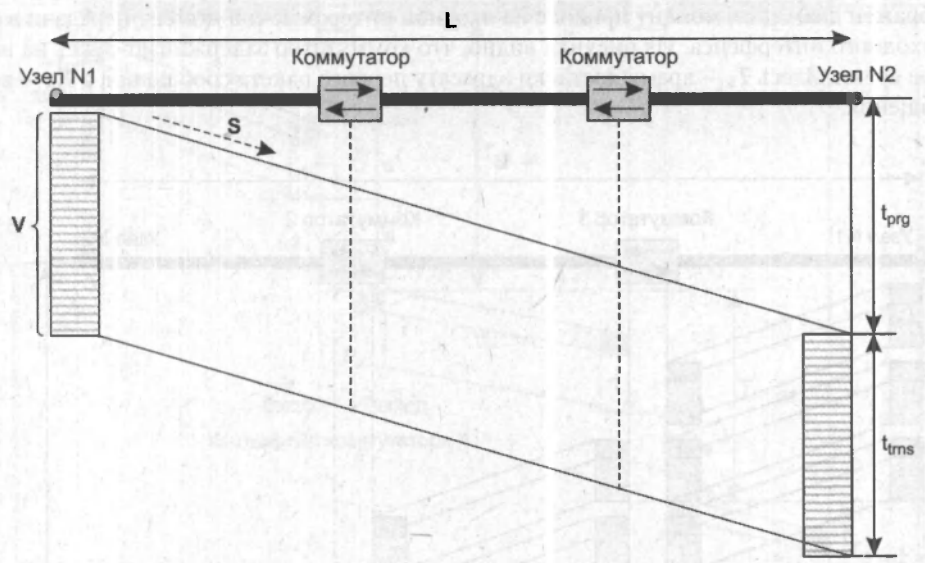


Рис. 3.12. Временная диаграмма передачи сообщения в сети с коммутацией каналов

В сети с коммутацией каналов данные после задержки, связанной с установлением канала, начинают передаваться на стандартной для канала скорости. Время доставки данных *T* адресату равно сумме *времени распространения сигнала в канале* t_{prg} и *времени передачи сообщения в канал* (называемое также *временем сериализации*) t_{trns}.

Наличие коммутаторов в сети с коммутацией каналов никак не влияет на суммарное время прохождения данных через сеть.

ПРИМЕЧАНИЕ

Заметим, что время передачи сообщения в канал в точности совпадает со временем приема сообщения из канала в буфер узла назначения, то есть временем буферизации.

Время распространения сигнала зависит от расстояния между абонентами L и скорости S распространения электромагнитных волн в конкретной физической среде, которая колеблется от 0,6 до 0,9 скорости света в вакууме:

$$t_{prg} = L/S.$$

Время передачи сообщения в канал (а значит, и время буферизации в узле назначения) равно отношению объема сообщения V в битах к пропускной способности канала C в битах в секунду:

$$t_{trna} = V/C.$$

В сети с коммутацией пакетов передача данных не требует обязательного установления соединения. Предположим, что в сеть, показанную на рис. 3.13, передается сообщение того же объема V, что и в предыдущем случае (см. рис. 3.12), однако оно разделено на пакеты, каждый из которых снабжен заголовком. Пакеты передаются от узла $N1$ узлу $N2$, между которыми расположены два коммутатора. На каждом коммутаторе каждый пакет изображен дважды: в момент прихода на входной интерфейс и в момент передачи в сеть с выходного интерфейса. Из рисунка видно, что коммутатор задерживает пакет на некоторое время. Здесь T_1 — время доставки адресату первого пакета сообщения, а T_{ps} — всего сообщения.

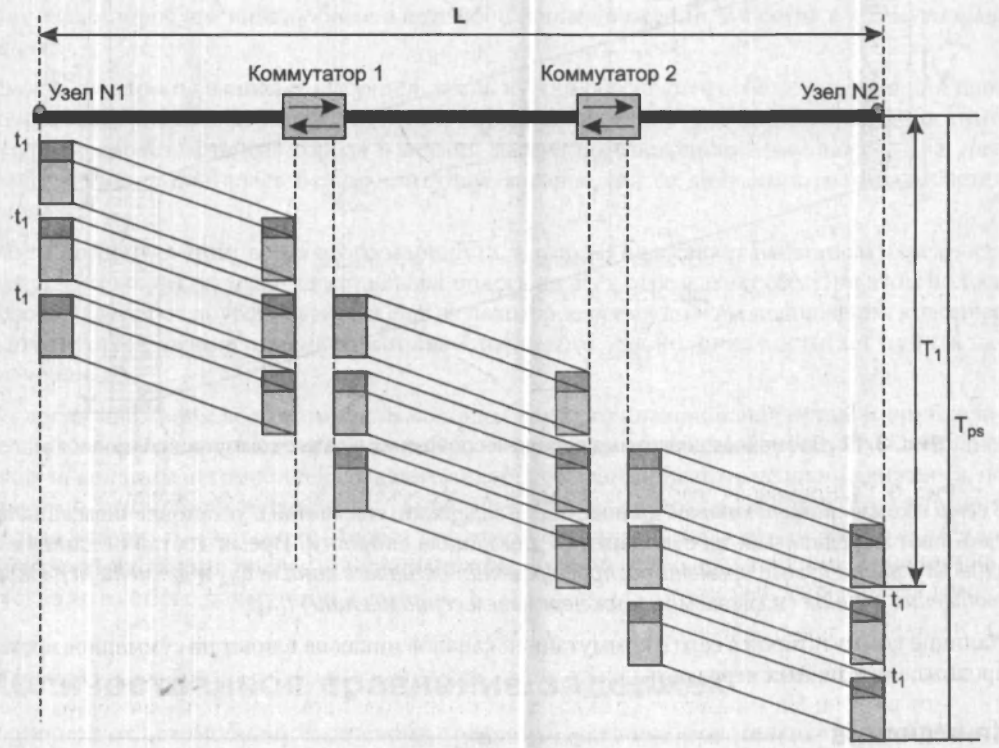


Рис. 3.13. Временна́я диаграмма передачи сообщения, разделенного на пакеты, в сети с коммутацией пакетов

Сравнивая временны́е диаграммы передачи данных в сетях с коммутацией каналов и пакетов, отметим два факта:

- значения времени распространения сигнала (t_{prg}) в одинаковой физической среде на одно и то же расстояние одинаковы;
- учитыва-я, что значения пропускной способности каналов в обеих сетях одинаковы, значения времени передачи сообщения в канал (t_{trns}) будут *также равны*.

Однако разбиение передаваемого сообщения на пакеты с последующей их передачей по сети с коммутацией пакетов приводит к дополнительным задержкам. Проследим путь первого пакета и отметим, из каких составляющих складывается время его передачи в узел назначения и какие из них специфичны для сети с коммутацией пакетов (рис. 3.14).

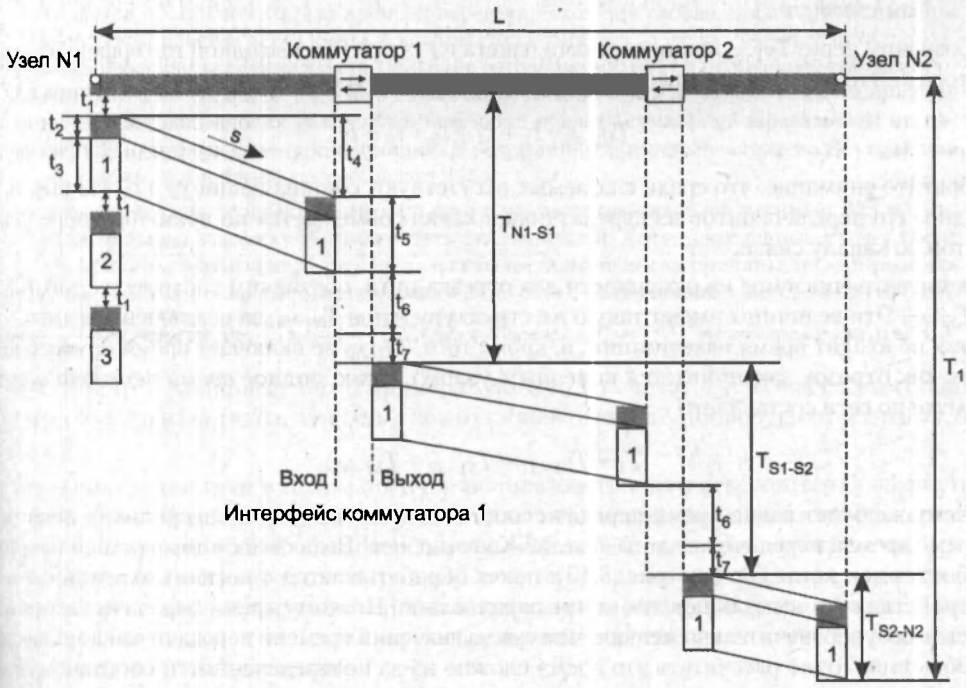


Рис. 3.14. Временная диаграмма передачи одного пакета в сети с коммутацией пакетов

Время передачи одного пакета от узла *N*1 до коммутатора *1* можно представить в виде суммы нескольких слагаемых.

- Во-первых, время тратится в узле-отправителе *N*1:
  - t_1 – время формирования пакета, также называемое временем пакетизации (зависит от различных параметров работы программного и аппаратного обеспечения узла-отправителя и не зависит от параметров сети);
  - t_2 – время передачи в канал заголовка;
  - t_3 – время передачи в канал поля данных пакета.

- Во-вторых, дополнительное время тратится на распространение сигналов по каналам связи. Обозначим через t_4 время распространения сигнала, представляющего один бит информации, от узла $N1$ до коммутатора *1*.
- В-третьих, дополнительное время тратится в промежуточном коммутаторе:
  - t_5 — время приема пакета с его заголовком из канала во входной буфер коммутатора; как уже было отмечено, это время равно ($t_2 + t_3$), то есть времени передачи пакета с заголовком в канал из узла источника;
  - t_6 — время ожидания пакета в очереди колеблется в очень широких пределах и заранее неизвестно, так как зависит от текущей загрузки сети;
  - t_7 — время коммутации пакета при его передаче в выходной порт фиксировано для конкретной модели и обычно невелико (от нескольких микросекунд до нескольких миллисекунд).

Обозначим через T_{N1-S1} время передачи пакета из узла $N1$ на выходной интерфейс коммутатора *1*. Это время складывается из следующих составляющих:

$$T_{N1-S1} = t_1 + t_4 + t_5 + t_6 + t_7.$$

Обратите внимание, что среди слагаемых отсутствуют составляющие t_2 и t_3. Из рис. 3.14 видно, что передача битов из передатчика в канал совмещается по времени с передачей битов по каналу связи.

Время, затрачиваемое на оставшиеся два отрезка пути, обозначим соответственно T_{S1-S2} и T_{S2-N2}. Эти величины имеют такую же структуру, что и T_{N1-S1}, за исключением того, что в них не входит время пакетизации, и, кроме того, T_{S2-N2} не включает время коммутации (так как отрезок заканчивается конечным узлом). Итак, полное *время передачи одного пакета* по сети составляет:

$$T_1 = T_{N1-S1} + T_{S1-S2} + T_{S2-N2}.$$

А чему же будет равно время передачи сообщения, состоящего из нескольких пакетов? Сумме времен передачи каждого пакета? Конечно, нет! Ведь сеть с коммутацией пакетов работает как конвейер (см. рис. 3.13): пакет обрабатывается в несколько этапов, и все устройства сети выполняют эти этапы параллельно. Поэтому время передачи такого сообщения будет значительно меньше, чем сумма значений времени передачи каждого пакета сообщения. Точно рассчитать это время сложно из-за неопределенности состояния сети, и вследствие этого, неопределенности значений времени ожидания пакетов в очередях коммутаторов. Однако если предположить, что пакеты стоят в очереди примерно одинаковое время, то общее *время передачи сообщения, состоящего из n пакетов*, можно оценить следующим образом:

$$T_{PS} = T_1 + (n - 1)(t_1 + t_5).$$

ПРИМЕР

Сравним задержки передачи данных в сетях с коммутацией пакетов с задержками в сетях с коммутацией каналов, основываясь на рис. 3.14. Пусть тестовое сообщение, которое нужно передать в обоих видах сетей, составляет 200 000 байт. Отправитель находится от получателя на расстоянии 5000 км. Пропускная способность линий связи составляет 2 Мбит/с. Время передачи данных по сети с коммутацией каналов складывается из времени распространения

сигнала, которое для расстояния 5000 км можно оценить примерно в 25 мс, и времени передачи сообщения в канал, которое при пропускной способности 2 Мбит/с и размере сообщения 200 000 байт равно примерно 800 мс, то есть всего передача данных абоненту занимает 825 мс. Оценим дополнительное время, которое требуется для передачи этого сообщения по сети с коммутацией пакетов. Будем считать, что путь от отправителя до получателя пролегает через 10 коммутаторов. Также предположим, что сеть работает в недогруженном режиме, то есть очереди в коммутаторах отсутствуют. Исходное сообщение разбивается на пакеты по 1000 байт, всего 200 пакетов.

Если принять интервал между отправкой пакетов равным 1 мс, тогда время передачи сообщения увеличится дополнительно на 200 мс. Время передачи сообщения в канал также увеличится из-за необходимости передавать заголовки пакетов. Предположим, что доля служебной информации, размещенной в заголовках пакетов, по отношению к общему объему сообщения составляет 10 %. Следовательно, дополнительная задержка, связанная с передачей заголовков пакетов, составляет 10 % от времени передачи исходного сообщения, то есть 80 мс. При прохождении пакетов через каждый коммутатор возникает задержка буферизации пакета. Эта задержка при величине пакета 1000 байт, заголовке 100 байт и пропускной способности линии 2 Мбит/с составляет 4,4 мс в одном коммутаторе. Плюс задержка коммутации 2 мс. В результате прохождения 10 коммутаторов пакет придет с суммарной задержкой 64 мс, потраченной на буферизацию и коммутацию. В результате дополнительная задержка, созданная сетью с коммутацией пакетов, составляет 344 мс.

Учитывая, что вся передача данных по сети с коммутацией каналов занимает 825 мс, эту дополнительную задержку можно считать существенной. Хотя приведенный расчет носит очень приблизительный характер, он делает более понятными те причины, по которым для отдельного абонента процесс передачи данных по сети с коммутацией пакетов является более медленным, чем по сети с коммутацией каналов.

Что же следует из приведенного примера? Можно ли считать, что сеть с коммутацией каналов более эффективна, чем сеть с коммутацией пакетов? Попробуем ответить на этот вопрос.

При рассмотрении сети в целом логично использовать в качестве критерия эффективности сети не скорость передачи трафика отдельного пользователя, а более интегральный критерий, например общий объем передаваемых сетью данных в единицу времени. В этом случае эффективность сетей с коммутацией пакетов по сравнению с сетями с коммутацией каналов (при равной пропускной способности каналов связи) оказывается выше. Такой результат был доказан в 60-е годы как экспериментально, так и аналитически с помощью теории массового обслуживания.

ПРИМЕР

Используем для сравнения эффективности сетей с коммутацией каналов и пакетов еще один пример (рис. 3.15). Два коммутатора объединены каналом связи с пропускной способностью 100 Мбит/с. Пользователи сети подключаются к сети с помощью каналов доступа (access link) с пропускной способностью 10 Мбит/с. Предположим, что все пользователи создают одинаковый пульсирующий трафик со средней скоростью 1 Мбит/с. При этом в течение непродолжительных периодов времени скорость данной предложенной нагрузки возрастает до максимальной скорости канала доступа, то есть до 10 Мбит/с. Такие периоды длятся не более одной секунды. Предположим также, что все пользователи, подключенные к коммутатору *S*1, передают информацию только пользователям, подключенным к коммутатору *S*2.

Пусть представленная на рисунке сеть является сетью с коммутацией каналов. Поскольку пики пользовательского трафика достигают 10 Мбит/с, каждому из пользователей необходимо

установить соединение с пропускной способностью 10 Мбит/с. Таким образом, одновременно через сеть смогут передавать данные только 10 пользователей. Суммарная средняя скорость передачи данных через сеть будет равна только 10 Мбит/с (10 пользователей передают данные со средней скоростью 1 Мбит/с). Следовательно, линия связи между коммутаторами, хотя и имеет общую пропускную способность 100 Мбит/с, используется только на 10 %.

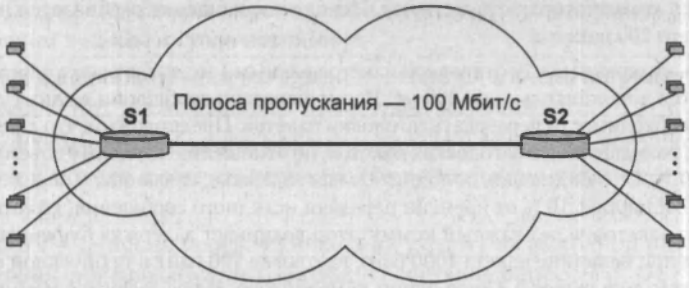


Рис. 3.15. Сравнение эффективности сетей с коммутацией пакетов и каналов

Теперь рассмотрим вариант, когда та же сеть работает на основе техники коммутации пакетов. При средней скорости пользовательских потоков 1 Мбит/с сеть может передавать одновременно до 100/1 = 100 (!) информационных потоков пользователей, полностью расходуя пропускную способность канала между коммутаторами. Однако это справедливо, если емкости буферов коммутаторов достаточно для хранения пакетов на периодах перегрузки, когда суммарная скорость потока данных превышает 100 Мбит/с. Оценим необходимый объем буфера коммутатора *S*1. За период перегрузки в коммутатор *S*1 от каждого потока поступит 10 Мбит/с × 1 с = 10 Мбит, а от 100 потоков — 1000 Мбит. Из этих данных за одну секунду коммутатор успеет передать в выходной канал только 100 Мбит. Значит, чтобы ни один пакет не был потерян во время перегрузки сети, общий объем входных буферов коммутатора должен быть не меньше 1000 – 100 = 900 Мбит, или более 100 Мбайт. Сегодняшние коммутаторы обычно имеют меньшие объемы буферов (1–10 Мбайт). Однако не нужно забывать, что вероятность совпадения периодов пиковой нагрузки у всех потоков, поступающих на входы коммутатора, очень мала. Так что даже если коммутатор имеет меньший объем буферной памяти, в подавляющем большинстве случаев он будет справляться с предложенной нагрузкой.

При сравнении сетей с коммутацией каналов и пакетов уместна аналогия с **мультипрограммными операционными системами**. Каждая отдельная программа в такой системе выполняется дольше, чем в однопрограммной системе, когда программе выделяется все процессорное время, пока она не завершит свое выполнение. Однако общее число программ, выполняемых в единицу времени, в мультипрограммной системе больше, чем в однопрограммной. Аналогично однопрограммной системе, в которой время от времени простаивает процессор или периферийные устройства, в сетях с коммутацией каналов при передаче пульсирующего трафика значительная часть зарезервированной пропускной способности каналов часто не используется.

Неопределенная пропускная способность сети с коммутацией пакетов — это плата за ее общую эффективность при некотором ущемлении интересов отдельных абонентов. Аналогично, в мультипрограммной операционной системе время выполнения приложения предсказать заранее невозможно, так как оно зависит от количества других приложений, с которыми делит процессор данное приложение.

В заключение этого раздела приведем табл. 3.1, в которой сведены свойства обоих видов сетей. На основании этих данных можно аргументировано утверждать, в каких случаях

рациональнее использовать сети с коммутацией каналов, а в каких — с коммутацией пакетов.

Таблица 3.1. Сравнение сетей с коммутацией каналов и пакетов

Коммутация каналов	Коммутация пакетов
Необходимо предварительно устанавливать соединение	Отсутствует этап установления соединения (дейтаграммный способ)
Адрес требуется только на этапе установления соединения	Адрес и другая служебная информация передаются с каждым пакетом
Сеть может отказать абоненту в установлении соединения	Сеть всегда готова принять данные от абонента
Гарантированная пропускная способность (полоса пропускания) для взаимодействующих абонентов	Пропускная способность сети для абонентов неизвестна, задержки передачи носят случайный характер
Трафик реального времени передается без задержек	Ресурсы сети используются эффективно при передаче пульсирующего трафика
Высокая надежность передачи	Возможные потери данных из-за переполнения буферов
Нерациональное использование пропускной способности каналов, снижающее общую эффективность сети	Автоматическое динамическое распределение пропускной способности физического канала между абонентами

Ethernet — пример стандартной технологии с коммутацией пакетов

Рассмотрим, каким образом описанные ранее концепции воплощены в одной из первых стандартных сетевых технологий — технологии Ethernet, работающей с битовой скоростью 10 Мбит/с. В этом разделе мы коснемся только самых общих принципов функционирования Ethernet. Детальное описание технологии Ethernet вы найдете в части III.

- *Топология*. Существует два варианта технологии Ethernet: Ethernet на разделяемой среде и коммутируемый вариант Ethernet. В первом случае все узлы сети разделяют общую среду передачи данных, и сеть строится по топологии общей шины. На рис. 3.16 показан простейший вариант топологии — все компьютеры сети подключены к общей разделяемой среде, состоящей из одного сегмента коаксиального кабеля.

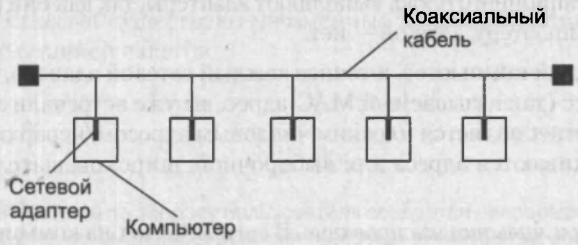


Рис. 3.16. Сеть Ethernet на разделяемой среде

В том случае, когда сеть Ethernet не использует разделяемую среду, а строится на коммутаторах, объединенных дуплексными каналами связи, говорят о коммутируемом

варианте Ethernet. Топология в этом случае является топологией дерева, то есть такой, при которой между двумя любыми узлами сеть существует ровно один путь. Пример топологии коммутируемой сети Ethernet показан на рис. 3.17.

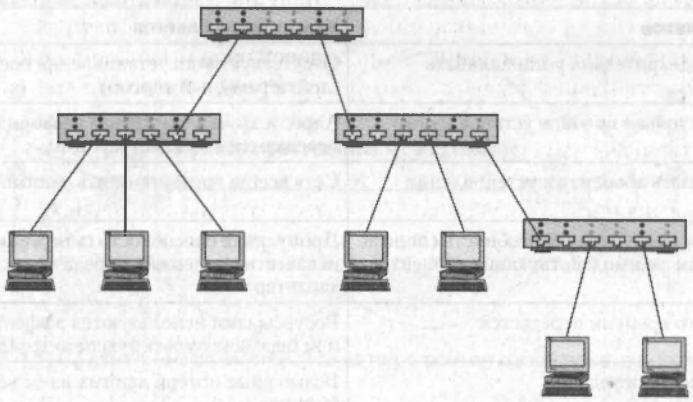


Рис. 3.17. Древовидная топология коммутируемой сети Ethernet

Топологические ограничения (только древовидная структура связей коммутаторов) связаны со способом построения таблиц продвижения коммутаторами Ethernet.

- *Способ коммутации.* В технологии Ethernet используется дейтаграммная коммутация пакетов. Единицы данных, которыми обмениваются компьютеры в сети Ethernet, называются кадрами. Кадр имеет фиксированный формат и наряду с полем данных содержит различную служебную информацию. В том случае, когда сеть Ethernet построена на коммутаторах, каждый коммутатор продвигает кадры в соответствии с теми принципами коммутации пакетов, которые были описаны ранее. А вот в случае односегментной сети Ethernet возникает законный вопрос: где же выполняется коммутация? Где хотя бы один коммутатор, который, как мы сказали, является главным элементом любой сети с коммутацией пакетов? Или же Ethernet поддерживает особый вид коммутации? Оказывается, коммутатор в односегментной сети Ethernet существует, но его не так просто разглядеть, потому что его функции распределены по всей сети. «Коммутатор» Ethernet состоит из сетевых адаптеров и разделяемой среды. Сетевые адаптеры представляют собой интерфейсы такого виртуального коммутатора, а разделяемая среда — коммутационный блок, который передает кадры между интерфейсами. Часть функций коммутационного блока выполняют адаптеры, так как они решают, какой кадр адресован их компьютеру, а какой — нет.
- *Адресация.* Каждый компьютер, а точнее каждый сетевой адаптер, имеет уникальный аппаратный адрес (так называемый MAC-адрес, вы уже встречали этот акроним в главе 2). Адрес Ethernet является плоским числовым адресом, иерархия здесь не используется. Поддерживаются адреса для выборочной, широковещательной и групповой рассылки.
- *Разделение среды и мультиплексирование.* В сети Ethernet на коммутаторах каждый канал является дуплексным каналом связи, и проблемы его разделения между интерфейсами узлов не возникает. Передатчики коммутаторов Ethernet используют дуплексные каналы связи для мультиплексирования потоков кадров от разных конечных узлов.

В случае Ethernet на разделяемой среде конечные узлы применяют специальный метод доступа с целью синхронизации использования единственного полудуплексного канала связи, объединяющего все компьютеры сети. Единого арбитра в сети Ethernet на разделяемой среде нет, вместо этого все узлы прибегают к распределенному случайному методу доступа. Информационные потоки, поступающие от конечных узлов сети Ethernet, мультиплексируются в единственном передающем канале в режиме разделения времени. То есть кадрам разных потоков поочередно предоставляется канал. Чтобы подчеркнуть не всегда очевидную разницу между понятиями мультиплексирования и разделения среды, рассмотрим ситуацию, когда из всех компьютеров сети Ethernet только одному нужно передавать данные, причем данные от нескольких приложений. В этом случае проблема разделения среды между сетевыми интерфейсами не возникает, в то время как задача передачи нескольких информационных потоков по общей линии связи (то есть мультиплексирование) остается.

- *Кодирование*. Адаптеры Ethernet работают с тактовой частотой 20 МГц, передавая в среду прямоугольные импульсы, соответствующие единицам и нулям данных компьютера. Когда начинается передача кадра, то все его биты передаются в сеть с постоянной скоростью 10 Мбит/с (каждый бит передается за два такта). Эта скорость определяется пропускной способностью линии связи в сети Ethernet.
- *Надежность*. Для повышения надежности передачи данных в Ethernet используется стандартный прием — подсчет **контрольной суммы** и передача ее в концевике кадра. Если принимающий адаптер путем повторного подсчета контрольной суммы обнаруживает ошибку в данных кадра, то такой кадр отбрасывается. Повторная передача кадра протоколом Ethernet не выполняется, эта задача должна решаться другими технологиями, например протоколом TCP в сетях TCP/IP.
- *Очереди*. В коммутируемых сетях Ethernet очереди кадров, готовых к отправке, организуются обычным для сетей с коммутацией пакетов способом, то есть с помощью буферной памяти интерфейсов коммутатора.

  В сетях Ethernet на разделяемой среде коммутаторы отсутствуют. На первый взгляд может показаться, что в Ethernet на разделяемой среде нет очередей, свойственных сетям с коммутацией пакетов. Однако отсутствие коммутатора с буферной памятью в сети Ethernet не означает, что очередей в ней нет. Просто здесь очереди переместились в буферную память сетевого адаптера. В те периоды времени, когда среда занята передачей кадров других сетевых адаптеров, данные (предложенная нагрузка) по-прежнему поступают в сетевой адаптер. Так как они не могут быть переданы в это время в сеть, они начинают накапливаться во внутреннем буфере адаптера Ethernet, образуя очередь. Поэтому в сети Ethernet существуют переменные задержки доставки кадров, как и во всех сетях с коммутацией пакетов.

Выводы

В сетях с коммутацией каналов по запросу пользователя создается непрерывный информационный канал, который образуется путем резервирования «цепочки» линий связи, соединяющих абонентов на время передачи данных. На всем своем протяжении канал передает данные с одной и той же скоростью. Это означает, что через сеть с коммутацией каналов можно качественно передавать данные, чувствительные к задержкам (голос, видео). Однако невозможность динамического перераспределения пропускной способности физического канала является принципиальным недостат-

ком сети с коммутацией каналов, который делает ее неэффективной для передачи пульсирующего компьютерного трафика.

При коммутации пакетов передаваемые данные разбиваются в исходном узле на небольшие части — пакеты. Пакет снабжается заголовком, в котором указывается адрес назначения, поэтому он может быть обработан коммутатором независимо от остальных данных. Коммутация пакетов повышает производительность сети при передаче пульсирующего трафика, так как при обслуживании большого числа независимых потоков периоды их активности не всегда совпадают во времени. Пакеты поступают в сеть без предварительного резервирования ресурсов в том темпе, в котором их генерирует источник. Однако этот способ коммутации имеет и отрицательные стороны: задержки передачи носят случайный характер, поэтому возникают проблемы при передаче трафика реального времени.

В сетях с коммутацией пакетов может использоваться один из трех алгоритмов продвижения пакетов: дейтаграммная передача, передача с установлением логического соединения и передача с установлением виртуального канала.

Вопросы и задания

1. Какие из приведенных утверждений верны при любых условиях:
   а) в сетях с коммутацией пакетов необходимо предварительно устанавливать соединение;
   б) в сетях с коммутацией каналов не требуется указывать адрес назначения данных;
   в) сеть с коммутацией пакетов более эффективна, чем сеть с коммутацией каналов;
   г) сеть с коммутацией каналов предоставляет взаимодействующим абонентам гарантированную пропускную способность.
2. Какие из сформулированных свойств составного канала всегда соответствуют действительности:
   а) данные, поступившие в составной канал, доставляются вызываемому абоненту без задержек и потерь;
   б) составной канал закрепляется за двумя абонентами на постоянной основе;
   в) количество элементарных каналов, входящих в составной канал между двумя абонентами, равно количеству промежуточных узлов плюс 1;
   г) составной канал имеет постоянную и фиксированную пропускную способность на всем своем протяжении.
3. Пусть для передачи голоса используется дискретизация по времени с интервалом 25 мкс и дискретизация по значениям на уровне 1024 градации звукового сигнала. Какая пропускная способность необходима для передачи полученного таким образом голосового трафика?
4. При каких условиях в коммутаторах сети с коммутацией пакетов должна быть предусмотрена буферизация? Варианты ответов:
   а) когда средняя скорость поступления данных в коммутатор превышает среднюю скорость их обработки коммутатором;
   б) всегда;
   в) если пакеты имеют большую длину;
   г) если пропускная способность сети ниже суммарной интенсивности источников трафика.

5. Определите, на сколько увеличится время передачи данных в сети с коммутацией пакетов по сравнению с сетью коммутации каналов, если известны следующие величины:
   - общий объем передаваемых данных — 200 Кбайт;
   - суммарная длина канала — 5000 км;
   - скорость передачи сигнала — 0,66 скорости света;
   - пропускная способность канала — 2 Мбит/с;
   - размер пакета без учета заголовка — 4 Кбайт;
   - размер заголовка — 40 байт;
   - интервал между пакетами — 1 мс;
   - количество промежуточных коммутаторов — 10;
   - время коммутации на каждом коммутаторе — 2 мс.

   Считайте, что сеть работает в недогруженном режиме, так что очереди в коммутаторах отсутствуют.

ГЛАВА 4 Архитектура и стандартизация сетей

Архитектура подразумевает представление сети в виде системы элементов, каждый из которых выполняет определенную частную функцию, при этом все элементы вместе согласованно решают общую задачу взаимодействия компьютеров. Другими словами, архитектура сети отражает декомпозицию общей задачи взаимодействия компьютеров на отдельные подзадачи, которые должны решаться отдельными элементами сети. Одним из важных элементов архитектуры сети является коммуникационный протокол — формализованный набор правил взаимодействия узлов сети.

Прорывом в стандартизации архитектуры компьютерной сети стала разработка модели взаимодействия открытых систем (Open System Interconnection, OSI), которая в начале 80-х годов обобщила накопленный к тому времени опыт. Модель OSI является международным стандартом и определяет способ декомпозиции задачи взаимодействия «по вертикали», поручая эту задачу коммуникационным протоколам семи уровней. Уровни образуют иерархию, известную как стек протоколов, где каждый вышестоящий уровень использует нижестоящий в качестве удобного инструмента для решения своих задач.

Существующие сегодня (или существовавшие еще недавно) стеки протоколов в целом отражают архитектуру модели OSI. Однако в каждом стеке протоколов имеются свои особенности и отличия от архитектуры OSI. Так, наиболее популярный стек TCP/IP состоит из четырех уровней. Стандартная архитектура компьютерной сети определяет также распределение протоколов между элементами сети — конечными узлами (компьютерами) и промежуточными узлами (коммутаторами и маршрутизаторами). Промежуточные узлы выполняют только транспортные функции стека протоколов, передавая трафик между конечными узлами. Конечные узлы поддерживают весь стек протоколов, предоставляя информационные услуги, например веб-сервис. Такое распределение функций означает смещение «интеллекта» сети на ее периферию.

Декомпозиция задачи сетевого взаимодействия

Организация взаимодействия между устройствами сети является сложной задачей. Для решения сложных задач используется известный универсальный прием — *декомпозиция*, то есть разбиение одной сложной задачи на несколько более простых задач-модулей. Декомпозиция состоит в четком определении функций каждого модуля, а также порядка их взаимодействия (то есть межмодульных интерфейсов). При таком подходе каждый модуль можно рассматривать как «черный ящик», абстрагируясь от его внутренних механизмов и концентрируя внимание на способе взаимодействия этих модулей. В результате такого логического упрощения задачи появляется возможность независимого тестирования, разработки и модификации модулей. Так, любой из показанных на рис. 4.1 модулей может быть переписан заново. Пусть, например, это будет модуль *A*, и если при этом разработчики сохранят без изменения межмодульные связи (в данном случае интерфейсы *A*-*B* и *A*-*C*), то это не потребует никаких изменений в остальных модулях.

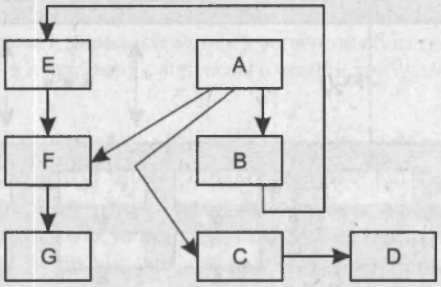


Рис. 4.1. Пример декомпозиции задачи

Многоуровневый подход

Еще более эффективной концепцией, развивающей идею декомпозиции, является *многоуровневый подход*. После представления исходной задачи в виде множества модулей эти модули группируют и упорядочивают по уровням, образующим иерархию. В соответствии с принципом иерархии для каждого промежуточного уровня можно указать непосредственно примыкающие к нему соседние вышележащий и нижележащий уровни (рис. 4.2).

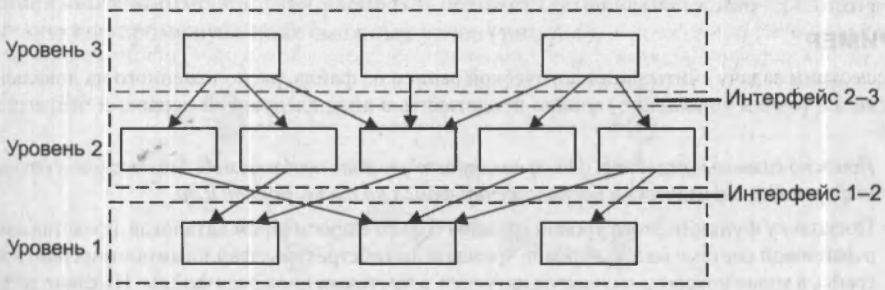


Рис. 4.2. Многоуровневый подход — создание иерархии задач

С одной стороны, группа модулей, составляющих каждый уровень, для решения своих задач должна обращаться с запросами только к модулям соседнего нижележащего уровня. С другой стороны, результаты работы каждого из модулей, отнесенных к некоторому уровню, могут быть переданы только модулям соседнего вышележащего уровня. Такая иерархическая декомпозиция задачи предполагает четкое определение функций и интерфейсов не только отдельных модулей, но и каждого уровня.

Межуровневый интерфейс, называемый также **интерфейсом услуг**, определяет набор функций, которые нижележащий уровень предоставляет вышележащему (рис. 4.3).

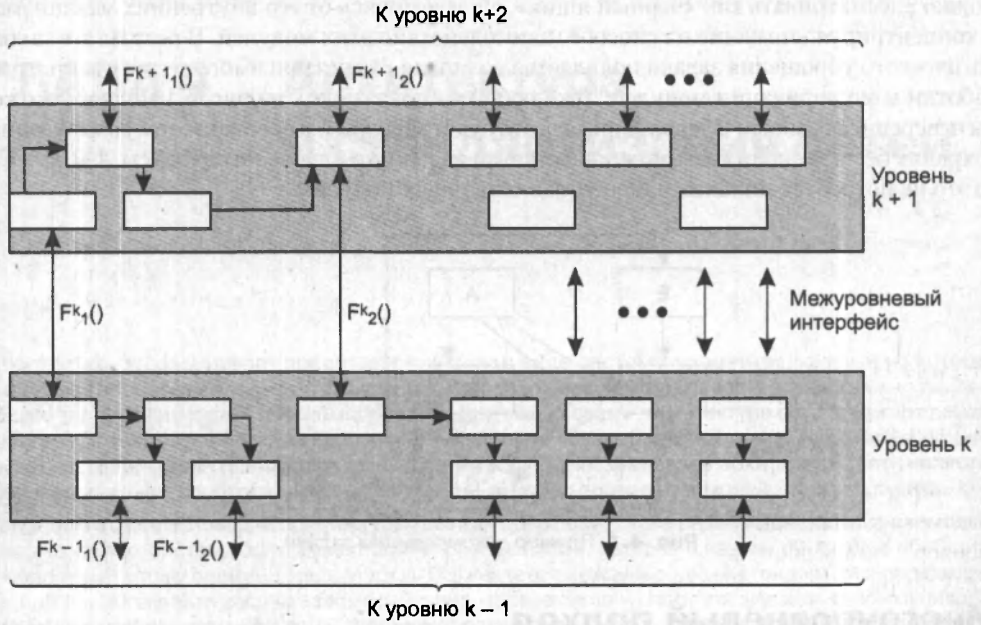


Рис. 4.3. Концепция многоуровневого взаимодействия

Такой подход дает возможность проводить разработку, тестирование и модификацию отдельного уровня независимо от других уровней. Иерархическая декомпозиция позволяет, двигаясь от более низкого уровня к более высокому, переходить ко все более и более абстрактному, а значит, более простому представлению исходной задачи.

ПРИМЕР

Рассмотрим задачу считывания логической записи из файла, расположенного на локальном диске. Ее (очень упрощенно) можно представить в виде следующей иерархии частных задач.

1. *Поиск по символьному имени файла его характеристик, необходимых для доступа к данным: информации о физическом расположении файла на диске, размер и др.*

   Поскольку функции этого уровня связаны только с просмотром каталогов, представления о файловой системе на этом уровне чрезвычайно абстрактны: файловая система имеет вид графа, в узлах которого находятся каталоги, а листьями являются файлы. Никакие детали физической и логической организации данных на диске данный уровень не интересуют.

2. *Определение считываемой части файла.*

   Для решения этой задачи необходимо снизить степень абстракции файловой системы. Функции данного уровня оперируют файлом как совокупностью определенным образом связанных физических блоков диска.

3. *Считывание данных с диска.*

   После определения номера физического блока файловая система обращается к системе ввода-вывода для выполнения операции чтения. На этом уровне уже фигурируют такие детали устройства файловой системы, как номера цилиндров, дорожек, секторов.

Среди функций, которые может запросить прикладная программа, обращаясь к верхнему уровню файловой системы, может быть, например, такая:

```
ПРОЧИТАТЬ 22 ЛОГИЧЕСКУЮ ЗАПИСЬ ФАЙЛА DIR1/MY/FILE.TXT
```

Верхний уровень не может выполнить этот запрос «только своими силами», определив по символьному имени DIR1/MY/FILE.TXT физический адрес файла, он обращается с запросом к нижележащему уровню:

```
ПРОЧИТАТЬ 22 ЛОГИЧЕСКУЮ ЗАПИСЬ ИЗ ФАЙЛА,
ИМЕЮЩЕГО ФИЗИЧЕСКИЙ АДРЕС 174 И РАЗМЕР 235
```

В ответ на запрос второй уровень определяет, что файл с адресом 174 занимает на диске пять несмежных областей, а искомая запись находится в четвертой области в физическом блоке 345. После этого он обращается к драйверу с запросом о чтении требуемой логической записи.

В соответствии с этой упрощенной схемой взаимодействие уровней файловой системы было однонаправленным — сверху вниз. Однако реальная картина существенно сложнее. Действительно, чтобы определить характеристики файла, верхний уровень должен «раскрутить» его символьное имя, то есть последовательно прочитать всю цепочку каталогов, указанную в имени файла. А это значит, что для решения свой задачи он несколько раз обратится к нижележащему уровню, который, в свою очередь, несколько раз «попросит» драйвер считать данные каталогов с диска. И каждый раз результаты будут передаваться снизу вверх.

Задача организации взаимодействия компьютеров в сети тоже может быть представлена в виде иерархически организованного множества модулей. Например, модулям нижнего уровня можно поручить вопросы, связанные с надежной передачей информации между двумя соседними узлами, а модулям следующего, более высокого уровня — транспортировку сообщений в пределах всей сети. Очевидно, что последняя задача — организация связи двух любых, не обязательно соседних, узлов — является более общей и поэтому ее решение может быть получено путем многократных обращений к нижележащему уровню. Так, организация взаимодействия узлов *A* и *B* может быть сведена к поочередному взаимодействию пар промежуточных смежных узлов (рис. 4.4).

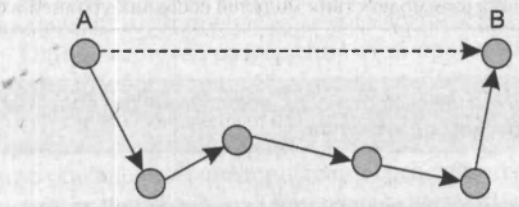


Рис. 4.4. Взаимодействие произвольной пары узлов

Протокол и стек протоколов

Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе обмена сообщениями участвуют, по меньшей мере, *две стороны*, то есть в данном случае необходимо организовать согласованную работу двух иерархий аппаратных и программных средств на разных компьютерах. Оба участника сетевого обмена должны принять множество соглашений. Например, они должны согласовать уровни и форму электрических сигналов, способ определения размера сообщений, договориться о методах контроля достоверности и т. п. Другими словами, соглашения должны быть приняты на всех уровнях, начиная от самого низкого — уровня передачи битов, и заканчивая самым высоким, реализующим обслуживание пользователей сети.

На рис. 4.5 показана модель взаимодействия двух узлов. С каждой стороны средства взаимодействия представлены четырьмя уровнями. Каждый уровень поддерживает интерфейсы двух типов. Во-первых, это интерфейсы услуг с выше- и нижележащим уровнями «своей» иерархии средств. Во-вторых, это интерфейс со средствами взаимодействия другой стороны, расположенными на том же уровне иерархии. Этот тип интерфейса называют **протоколом**. Таким образом, протокол всегда является одноранговым интерфейсом.

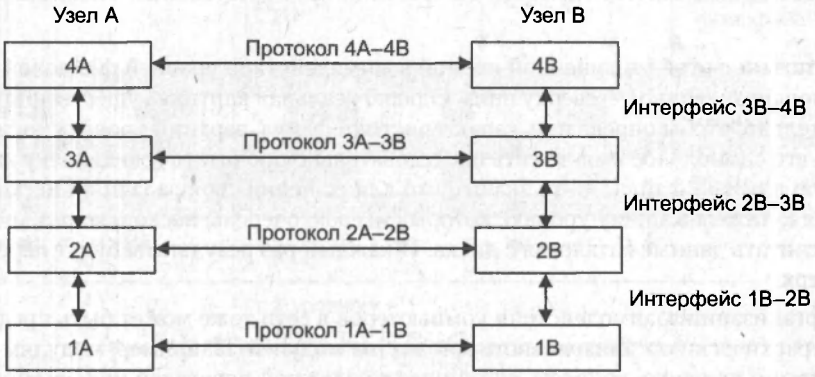


Рис. 4.5. Взаимодействие двух узлов

ПРИМЕЧАНИЕ

В сущности, термины «протокол» и «интерфейс» выражают одно и то же понятие — *формализованное описание процедуры взаимодействия двух объектов*, но традиционно в сетях за ними закрепили разные области действия: протоколы определяют правила взаимодействия модулей одного уровня в разных узлах, а интерфейсы — правила взаимодействия модулей соседних уровней в одном узле.

Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети, называется **стеком протоколов**.

Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней, как правило, программными средствами.

Программный модуль, реализующий некоторый протокол, называют протокольной сущностью, или, для краткости, тоже протоколом. Понятно, что один и тот же протокол может быть реализован с разной степенью эффективности. Именно поэтому при сравнении протоколов следует учитывать не только логику их работы, но и качество программной реализации. Более того, на эффективность взаимодействия устройств в сети влияет качество всей совокупности протоколов, составляющих стек, в частности то, *насколько рационально распределены функции между протоколами* разных уровней и насколько хорошо определены интерфейсы между ними.

Протокольные сущности одного уровня двух взаимодействующих сторон обмениваются сообщениями в соответствии с определенным для них протоколом. Сообщения состоят из заголовка и поля данных (иногда оно может отсутствовать). Обмен сообщениями является своеобразным языком общения, с помощью которого каждая из сторон «объясняет» другой стороне, что необходимо сделать на каждом этапе взаимодействия. Работа каждого протокольного модуля состоит в интерпретации заголовков поступающих к нему сообщений и выполнении связанных с этим действий. Заголовки сообщений разных протоколов имеют разную структуру, что соответствует различиям в их функциональности. Понятно, что чем сложнее структура заголовка сообщения, тем более сложные функции возложены на соответствующий протокол.

Модель OSI

Из того что протокол является соглашением, принятым двумя взаимодействующими узлами сети, совсем не следует, что он обязательно является стандартным. Но на практике при реализации сетей стремятся использовать стандартные протоколы. Это могут быть фирменные, национальные или международные стандарты.

В начале 80-х годов ряд международных организаций по стандартизации, в частности International Organization for Standardization (ISO), часто называемая International Standards Organization, а также International Telecommunications Union (ITU) и некоторые другие, разработали стандартную модель **взаимодействия открытых систем** (Open System Interconnection, OSI). Эта модель сыграла значительную роль в развитии компьютерных сетей.

Общая характеристика модели OSI

К концу 70-х годов в мире уже существовало большое количество фирменных стеков коммуникационных протоколов, среди которых можно назвать, например, такие популярные стеки, как DECnet, TCP/IP и SNA. Подобное разнообразие средств межсетевого взаимодействия вывело на первый план проблему несовместимости устройств, использующих разные протоколы. Одним из путей разрешения этой проблемы в то время виделся всеобщий переход на единый, общий для всех систем стек протоколов, созданный с учетом недостатков уже существующих стеков. Такой академический подход к созданию нового стека начался с разработки модели OSI и занял семь лет (с 1977 по 1984 год). Назначение модели OSI состоит в обобщенном представлении средств сетевого взаимодействия. Она разрабатывалась в качестве своего рода универсального языка сетевых специалистов, именно поэтому ее называют справочной моделью.

ВНИМАНИЕ

Модель OSI определяет, во-первых, уровни взаимодействия систем в сетях с коммутацией пакетов, во-вторых, стандартные названия уровней, в-третьих, функции, которые должен выполнять каждый уровень. Модель OSI не содержит описаний реализаций конкретного набора протоколов.

В модели OSI средства взаимодействия делятся на *семь* уровней: прикладной, представления, сеансовый, транспортный, сетевой, канальный и физический (рис. 4.6). Каждый уровень имеет дело с совершенно определенным аспектом взаимодействия сетевых устройств.

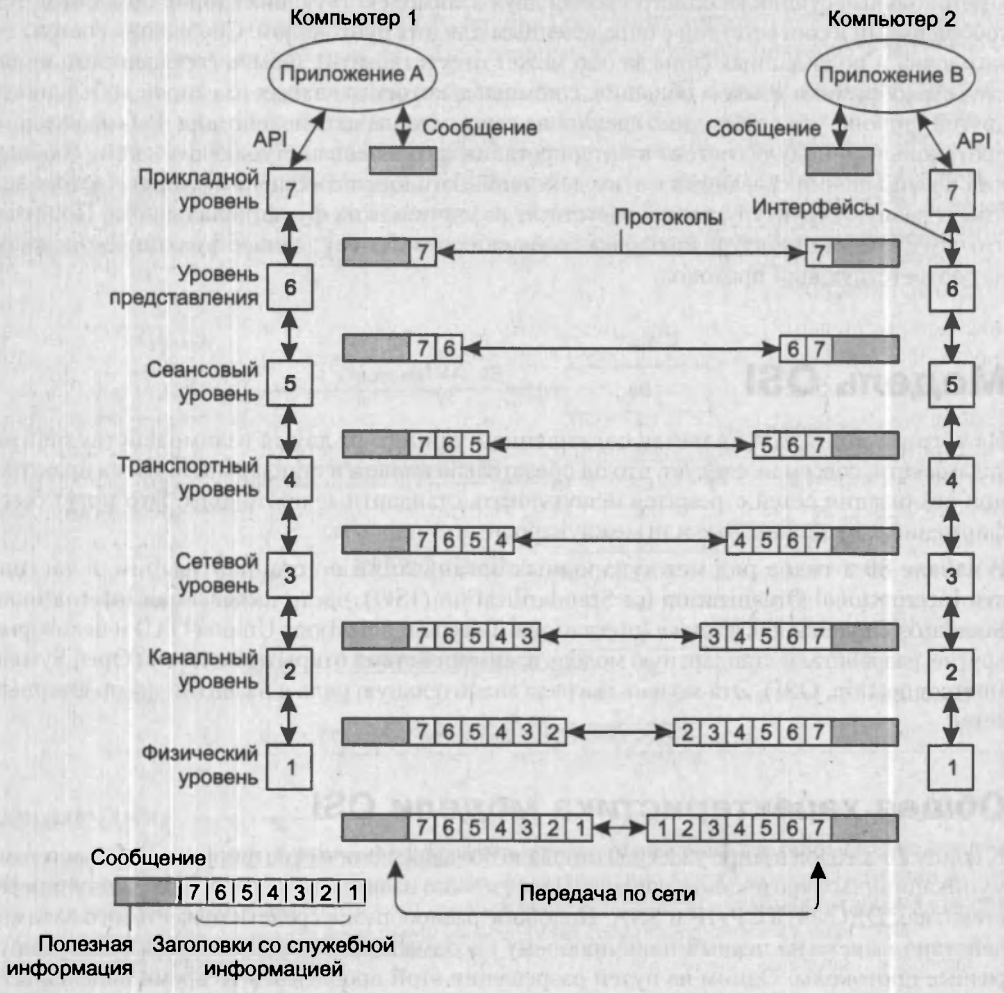


Рис. 4.6. Модель взаимодействия открытых систем ISO/OSI

ВНИМАНИЕ

Модель OSI описывает только системные средства взаимодействия, реализуемые операционной системой, системными утилитами, системными аппаратными средствами. Модель не включает средства взаимодействия приложений конечных пользователей. Важно различать уровень взаимодействия приложений и прикладной уровень семиуровневой модели.

Приложения могут реализовывать собственные протоколы взаимодействия, используя для этих целей многоуровневую совокупность системных средств. Именно для этого в распоряжение программистов предоставляется **прикладной программный интерфейс** (Application Program Interface, API). В соответствии с идеальной схемой модели OSI приложение может обращаться с запросами только к самому верхнему уровню — прикладному, однако на практике многие стеки коммуникационных протоколов предоставляют возможность программистам напрямую обращаться к сервисам, или службам, расположенных ниже уровней.

Например, некоторые СУБД имеют встроенные средства удаленного доступа к файлам. В этом случае приложение, выполняя доступ к удаленным ресурсам, не использует системную файловую службу; оно обходит верхние уровни модели OSI и обращается непосредственно к ответственным за транспортировку сообщений по сети системным средствам, которые располагаются на нижних уровнях модели OSI.

Итак, пусть приложение узла *A* хочет взаимодействовать с приложением узла *B*. Для этого приложение *A* обращается с запросом к прикладному уровню, например к файловой службе. На основании этого запроса программное обеспечение прикладного уровня формирует сообщение стандартного формата. Но для того чтобы доставить эту информацию по назначению, предстоит решить еще много задач, ответственность за которые несут нижележащие уровни.

После формирования сообщения прикладной уровень направляет его вниз по стеку уровню представления. Протокол уровня представления на основании информации, полученной из заголовка сообщения прикладного уровня, выполняет требуемые действия и добавляет к сообщению собственную служебную информацию — заголовок уровня представления, в котором содержатся указания для протокола уровня представления машины-адресата. Полученное в результате сообщение передается вниз сеансовому уровню, который, в свою очередь, добавляет свой заголовок и т. д. (Некоторые реализации протоколов помещают служебную информацию не только в начале сообщения в виде заголовка, но и в конце в виде так называемого концевика.) Наконец, сообщение достигает нижнего, физического, уровня, который собственно и передает его по линиям связи машине-адресату. К этому моменту сообщение «обрастает» заголовками всех уровней (рис. 4.7).

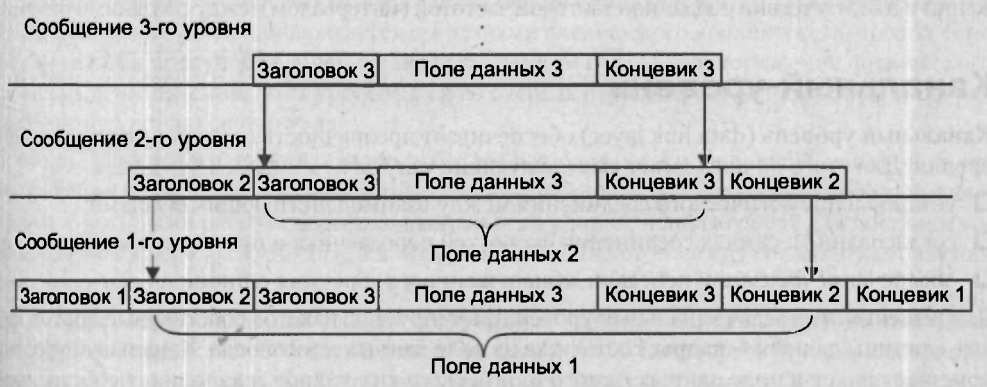


Рис. 4.7. Вложенность сообщений различных уровней

Физический уровень помещает сообщение на физический выходной интерфейс компьютера 1, и оно начинает свое «путешествие» по сети (до этого момента сообщение передавалось от одного уровню другому в пределах компьютера 1).

Когда сообщение но сети поступает на входной интерфейс компьютера 2, оно принимается его физическим уровнем и последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует и обрабатывает заголовок своего уровня, выполняя соответствующие функции, а затем удаляет этот заголовок и передает сообщение вышележащему уровню.

Как видно из описания, протокольные сущности одного уровня не общаются между собой непосредственно, в этом общении всегда участвуют посредники — средства протоколов нижележащих уровней. И только физические уровни различных узлов взаимодействуют непосредственно.

В стандартах ISO для обозначения единиц обмена данными, с которыми имеют дело протоколы разных уровней, используется общее название **протокольная единица данных** (Protocol Data Unit, PDU). Для обозначения единиц обмена данными конкретных уровней часто используются *специальные* названия, в частности: **сообщение**, **кадр**, **пакет**, **дейтаграмма**, **сегмент**.

Физический уровень

Физический уровень (physical layer) имеет дело с передачей потока битов по физическим каналам связи, таким как коаксиальный кабель, витая пара, оптоволоконный кабель или цифровой территориальный канал. Функции физического уровня реализуются на всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом.

Примером протокола физического уровня может служить спецификация 10Base-T технологии Ethernet, которая определяет в качестве используемого кабеля неэкранированную витую пару категории 3 с волновым сопротивлением 100 Ом, разъем RJ-45, максимальную длину физического сегмента 100 метров, манчестерский код для представления данных в кабеле, а также некоторые другие характеристики среды и электрических сигналов.

Физический уровень не вникает в смысл информации, которую он передает. Для него эта информация представляет собой однородный поток битов, которые нужно доставить без искажений и в соответствии с заданной тактовой частотой (интервалом между соседними битами).

Канальный уровень

Канальный уровень (data link layer) обеспечивает прозрачность соединения для сетевого уровня. Для этого он предлагает ему следующие услуги:

- установление логического соединения между взаимодействующими узлами;
- согласование в рамках соединения скоростей передатчика и приемника информации;
- обеспечение надежной передачи, обнаружение и коррекция ошибок.

Для решения этих задач канальный уровень формирует из пакетов собственные протокольные единицы данных — **кадры**, состоящие из поля данных и заголовка. Канальный уровень помещает пакет в поле данных одного или нескольких кадров и заполняет собственной служебной информацией заголовок кадра.

В сетях, построенных на основе разделяемой среды, физический уровень выполняет еще одну функцию — проверяет доступность разделяемой среды. Эту функцию иногда выделяют в отдельный подуровень **управления доступом к среде** (Medium Access Control, MAC).

Протоколы канального уровня реализуются как на конечных узлах (средствами сетевых адаптеров и их драйверов), так и на всех промежуточных сетевых устройствах.

Рассмотрим более подробно работу канального уровня, начиная с момента, когда сетевой уровень отправителя передает канальному уровню пакет, а также указание, какому узлу его передать. Для решения этой задачи канальный уровень создает кадр, который имеет поле данных и заголовок. Канальный уровень помещает (*инкапсулирует*) пакет в поле данных кадра и заполняет соответствующей служебной информацией заголовок кадра. Важнейшей информацией заголовка кадра является адрес назначения, на основании которого коммутаторы сети будут продвигать пакет.

Одной из задач канального уровня является *обнаружение и коррекция ошибок*. Канальный уровень может обеспечить надежность передачи, например, путем фиксирования границ кадра, помещая специальную последовательность битов в его начало и конец, а затем добавляя к кадру контрольную сумму. Контрольная сумма вычисляется по некоторому алгоритму как функция от всех байтов кадра. На стороне получателя канальный уровень группирует биты, поступающие с физического уровня, в кадры, снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой, переданной в кадре. Если они совпадают, кадр считается правильным. Если же контрольные суммы не совпадают, фиксируется ошибка.

В функции канального уровня входит не только обнаружение ошибок, но и их исправление за счет *повторной передачи поврежденных кадров*. Однако эта функция не является обязательной и в некоторых реализациях канального уровня она отсутствует, например, в Ethernet.

Прежде чем переправить кадр физическому уровню для непосредственной передачи данных в сеть, канальному уровню может потребоваться решить еще одну важную задачу. Если в сети используется разделяемая среда, то прежде чем физический уровень начнет передавать данные, канальный уровень должен *проверить доступность среды*. Как уже отмечалось, функции проверки доступности разделяемой среды иногда выделяют в отдельный подуровень управления доступом к среде (подуровень MAC).

Если разделяемая среда освободилась (когда она не используется, то такая проверка, конечно, пропускается), кадр передается средствами физического уровня в сеть, проходит по каналу связи и поступает в виде последовательности битов в распоряжение физического уровня узла назначения. Этот уровень в свою очередь передает полученные биты «наверх» канальному уровню своего узла.

Протокол канального уровня обычно работает в пределах сети, являющейся одной из составляющих более крупной составной сети, объединенной протоколами сетевого уровня. Адреса, с которыми работает протокол канального уровня, используются для доставки кадров только в пределах этой сети, а для перемещения пакетов между сетями применяются уже адреса следующего, сетевого, уровня.

В локальных сетях канальный уровень поддерживает весьма мощный и законченный набор функций по пересылке сообщений между узлами сети. В некоторых случаях протоколы канального уровня локальных сетей оказываются самодостаточными транспортными средствами и могут допускать работу непосредственно поверх себя протоколов прикладного уровня или приложений без привлечения средств сетевого и транспортного уровней. Тем не менее для качественной передачи сообщений в сетях с произвольной топологией функций канального уровня оказывается недостаточно.

Сетевой уровень

Сетевой уровень (network layer) служит для образования единой транспортной системы, объединяющей несколько сетей, называемой **составной сетью**, или **интернетом**[1].

Технология, позволяющая соединять в единую сеть множество сетей, в общем случае построенных на основе разных технологий, называется технологией **межсетевого взаимодействия** (internetworking).

На рис. 4.8 показано несколько сетей, каждая из которых использует собственную технологию канального уровня: Ethernet, FDDI, Token Ring, ATM, Frame Relay. На базе этих технологий любая из указанных сетей может связывать между собой любых пользователей, но только *своей* сети, и не способна обеспечить передачу данных в другую сеть. Причина такого положения вещей очевидна и кроется в существенных отличиях одной технологии от другой. Даже наиболее близкие технологии LAN — Ethernet, FDDI, Token Ring, — имеющие одну и ту же систему адресации (адреса подуровня MAC, называемые MAC-адресами), отличаются друг от друга форматом используемых кадров и логикой работы протоколов. Еще больше отличий между технологиями LAN и WAN. Во многих технологиях WAN задействована техника предварительно устанавливаемых виртуальных каналов, идентификаторы которых применяются в качестве адресов. Все технологии имеют собственные форматы кадров (в технологии ATM кадр даже называется иначе — ячейкой) и, конечно, собственные стеки протоколов.

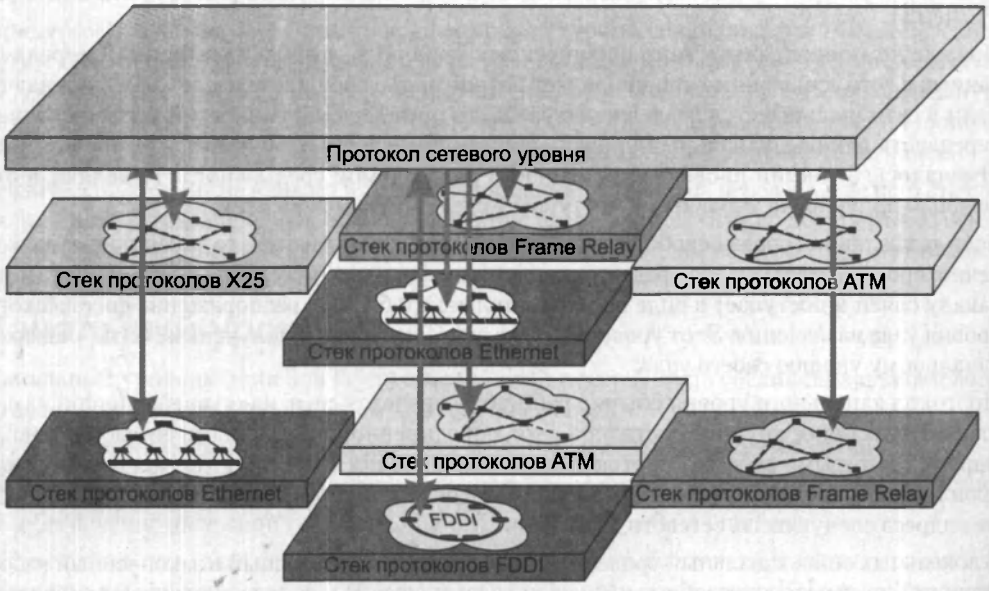


Рис. 4.8. Необходимость сетевого уровня

[1] Не следует путать интернет (со строчной буквы) с Интернетом (с прописной буквы). Интернет – это самая известная и охватывающая весь мир реализация составной сети, построенная на основе технологии TCP/IP.

Чтобы связать между собой сети, построенные на основе столь отличающихся технологий, нужны *дополнительные средства*, и такие средства предоставляет сетевой уровень.

Функции сетевого уровня реализуются:

- группой протоколов;
- специальными устройствами — **маршрутизаторами**.

Одной из функций маршрутизатора является *физическое соединение сетей*. Маршрутизатор имеет несколько сетевых интерфейсов, подобных интерфейсам компьютера, к каждому из которых может быть подключена одна сеть. Таким образом, все интерфейсы маршрутизатора можно считать узлами разных сетей. Маршрутизатор может быть реализован программно на базе универсального компьютера (например, типовая конфигурация Unix или Windows включает программный модуль маршрутизатора). Однако чаще маршрутизаторы реализуются на базе специализированных аппаратных платформ. В состав программного обеспечения маршрутизатора входят протокольные модули сетевого уровня.

Итак, чтобы связать сети, показанные на рис. 4.8, необходимо соединить все эти сети маршрутизаторами и установить протокольные модули сетевого уровня на все конечные узлы пользователей, которые хотели бы связываться через составную сеть (рис. 4.9).

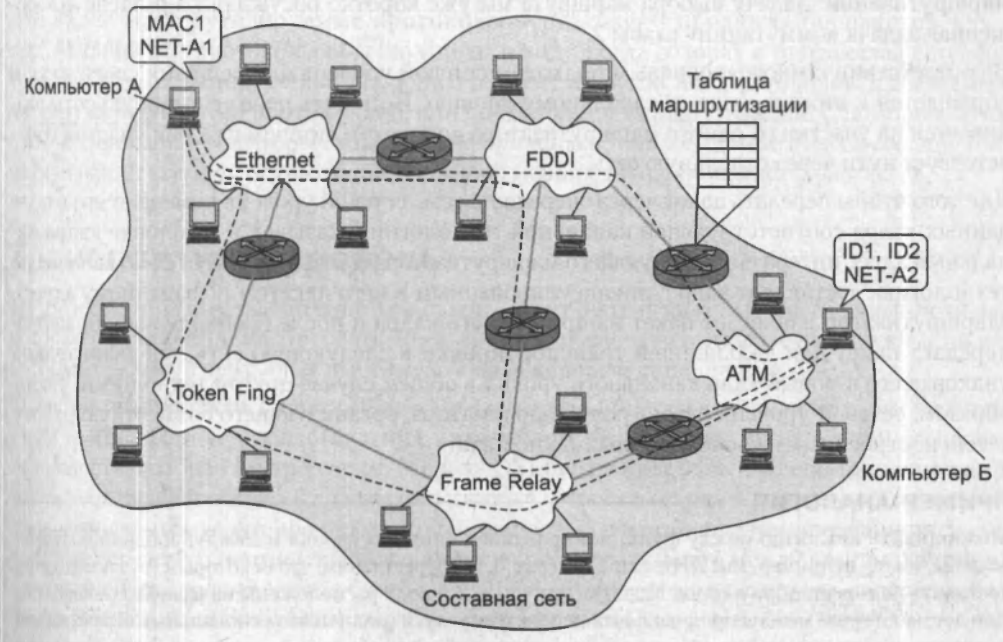


Рис. 4.9. Пример составной сети

Данные, которые необходимо передать через составную сеть, поступают на сетевой уровень от вышележащего транспортного уровня. Эти данные снабжаются заголовком сетевого уровня. Данные вместе с заголовком образуют **пакет** — так называется PDU сетевого уровня. Заголовок пакета сетевого уровня имеет унифицированный формат, не зависящий от форматов кадров канального уровня тех сетей, которые могут входить в составную сеть,

и несет, наряду с другой служебной информацией, данные об адресе назначения этого пакета.

Для того чтобы протоколы сетевого уровня могли доставлять пакеты любому узлу составной сети, эти узлы должны иметь адреса, уникальные в пределах данной составной сети. Такие адреса называются **сетевыми**, или **глобальными**. Каждый узел составной сети, который намерен обмениваться данными с другими узлами составной сети, наряду с адресом, назначенным ему на канальном уровне, должен иметь сетевой адрес. Например, на рис. 4.9 компьютер в сети Ethernet, входящей в составную сеть, имеет адрес канального уровня MAC1 и адрес сетевого уровня NET-A1; аналогично в сети ATM узел, адресуемый идентификаторами виртуальных каналов ID1 и ID2, имеет сетевой адрес NET-A2. В пакете в качестве адреса назначения должен быть указан адрес сетевого уровня, на основании которого определяется маршрут пакета.

Определение маршрута является важной задачей сетевого уровня. **Маршрут** описывается последовательностью сетей (или маршрутизаторов), через которые должен пройти пакет, чтобы попасть к адресату. Например, на рис. 4.9 штриховой линией показано три маршрута, по которым могут быть переданы данные от компьютера *А* к компьютеру *Б*. Маршрутизатор собирает информацию о топологии связей между сетями и на основе этой информации строит таблицы коммутации, которые в данном случае носят специальное название **таблиц маршрутизации**. Задачу выбора маршрута мы уже коротко обсуждали в разделе «Обобщенная задача коммутации» главы 2.

В соответствии с многоуровневым подходом сетевой уровень для решения своей задачи обращается к нижележащему канальному уровню. Весь путь через составную сеть разбивается на участки от одного маршрутизатора до другого, причем каждый участок соответствует пути через отдельную сеть.

Для того чтобы передать пакет через очередную сеть, сетевой уровень помещает его в поле данных кадра соответствующей канальной технологии, указывая в заголовке кадра канальный адрес интерфейса следующего маршрутизатора. Сеть, используя свою канальную технологию, доставляет кадр с инкапсулированным в него пакетом по заданному адресу. Маршрутизатор извлекает пакет из прибывшего кадра и после необходимой обработки передает пакет для дальнейшей транспортировки в следующую сеть, предварительно упаковав его в новый кадр канального уровня в общем случае другой технологии. Таким образом, сетевой уровень играет роль координатора, организующего совместную работу сетей, построенных на основе разных технологий.

ПРИМЕР-АНАЛОГИЯ

Можно найти аналогию между функционированием сетевого уровня и международной почтовой службы, такой, например, как DHL или TNT (рис. 4.10). Представим, что некоторый груз необходимо доставить из города Абра в город Кадабра, причем эти города расположены на разных континентах. Для доставки груза международная почта использует услуги различных региональных перевозчиков: железную дорогу, морской транспорт, авиаперевозчиков, автомобильный транспорт. Эти перевозчики могут рассматриваться как аналоги сетей канального уровня, причем каждая «сеть» здесь построена на основе собственной технологии. Из этих региональных служб международная почтовая служба должна организовать единую слаженно работающую сеть. Для этого международная почтовая служба должна, во-первых, продумать маршрут перемещения почты, во-вторых, координировать работу в пунктах смены перевозчиков (например, выгружать почту из вагонов и размещать ее в транспортном отсеке самолета). Каждый же перевозчик ответственен только за перемещение почты по своей части пути и не несет никакой ответственности за состояние почты за его пределами.

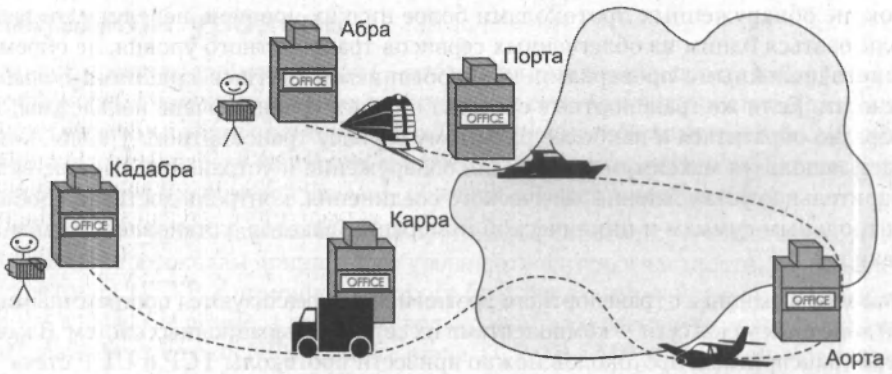


Рис. 4.10. Работа международной почтовой службы

В общем случае функции сетевого уровня шире, чем обеспечение обмена в пределах составной сети. Так, сетевой уровень решает задачу создания надежных и гибких барьеров на пути нежелательного трафика между сетями.

В заключение отметим, что на сетевом уровне определяются два вида протоколов. Первый вид — **маршрутизируемые протоколы** — реализуют продвижение пакетов через сеть. Именно эти протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня. Однако часто к сетевому уровню относят и другой вид протоколов, называемых **маршрутизирующими протоколами**, или **протоколами маршрутизации**. С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений, на основании которой осуществляется выбор маршрута продвижения пакетов.

Транспортный уровень

На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением.

Транспортный уровень (transport layer) обеспечивает приложениям или верхним уровням стека — прикладному, представления и сеансовому — передачу данных с той степенью надежности, которая им требуется. Модель OSI определяет пять **классов транспортного сервиса** от низшего класса 0 до высшего класса 4. Эти виды сервиса отличаются качеством предоставляемых услуг: срочностью, возможностью восстановления прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, а главное — способностью к обнаружению и исправлению ошибок передачи, таких как искажение, потеря и дублирование пакетов.

Выбор класса сервиса транспортного уровня определяется, с одной стороны, тем, в какой степени задача обеспечения надежности решается самими приложениями и протоколами более высоких, чем транспортный, уровней. С другой стороны, этот выбор зависит от того, насколько надежной является система транспортировки данных в сети, обеспечиваемая уровнями, расположенными ниже транспортного: сетевым, канальным и физическим. Так, если качество каналов передачи связи очень высокое и вероятность возникновения

ошибок, не обнаруженных протоколами более низких уровней, невелика, то разумно воспользоваться одним из облегченных сервисов транспортного уровня, не обремененных многочисленными проверками, квитированием и другими приемами повышения надежности. Если же транспортные средства нижних уровней очень ненадежны, то целесообразно обратиться к наиболее развитому сервису транспортного уровня, который работает, используя максимум средств для обнаружения и устранения ошибок, включая предварительное установление логического соединения, контроль доставки сообщений по контрольным суммам и циклической нумерации пакетов, установление тайм-аутов доставки и т. п.

Все протоколы, начиная с транспортного уровня и выше, реализуются программными средствами конечных узлов сети — компонентами их сетевых операционных систем. В качестве примера транспортных протоколов можно привести протоколы TCP и UDP стека TCP/IP и протокол SPX стека Novell.

Протоколы нижних четырех уровней обобщенно называют сетевым транспортом, или транспортной подсистемой, так как они полностью решают задачу транспортировки сообщений с заданным уровнем качества в составных сетях с произвольной топологией и различными технологиями. Оставшиеся три верхних уровня решают задачи предоставления прикладных сервисов, используя нижележащую транспортную подсистему.

Сеансовый уровень

Сеансовый уровень (session layer) управляет взаимодействием сторон: фиксирует, какая из сторон является активной в настоящий момент, и предоставляет средства синхронизации сеанса. Эти средства позволяют в ходе длинных передач сохранять информацию о состоянии этих передач в виде контрольных точек, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, а не начинать все с начала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется в виде отдельных протоколов. Функции этого уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе.

Уровень представления

Уровень представления (presentation layer), как явствует из его названия, обеспечивает представление передаваемой по сети информации, не меняя при этом ее содержания. За счет уровня представления информация, передаваемая прикладным уровнем одной системы, всегда понятна прикладному уровню другой системы. С помощью средств данного уровня протоколы прикладных уровней могут преодолеть синтаксические различия в представлении данных или же различия в кодах символов, например кодов ASCII и EBCDIC. На этом уровне могут выполняться шифрование и дешифрирование данных, благодаря которым секретность обмена данными обеспечивается сразу для всех прикладных служб. Примером такого протокола является протокол SSL (Secure Socket Layer — слой защищенных сокетов), который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня стека TCP/IP.

Прикладной уровень

Прикладной уровень (application layer) — это в действительности просто набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые веб-страницы, а также организуют свою совместную работу, например, по протоколу электронной почты. Единица данных, которой оперирует прикладной уровень, обычно называется **сообщением**.

Существует очень большое разнообразие протоколов и соответствующих служб прикладного уровня. К протоколам прикладного уровня относится, в частности, упоминавшийся ранее протокол HTTP, с помощью которого браузер взаимодействует с веб-сервером. Приведем в качестве примера также несколько наиболее распространенных реализаций сетевых файловых служб: NFS и FTP в стеке TCP/IP, SMB в Microsoft Windows, NCP в операционной системе Novell NetWare.

Модель OSI и сети с коммутацией каналов

Как уже было упомянуто, модель OSI описывает процесс взаимодействия устройств в сети с **коммутацией пакетов**. А как же обстоит дело с сетями **коммутации каналов**? Существует ли для них собственная справочная модель? Можно ли сопоставить функции технологий коммутации каналов с уровнями модели OSI?

Да, для представления структуры средств межсетевого взаимодействия сетей с коммутацией каналов также используется многоуровневый подход, в соответствии с которым существуют протоколы нескольких уровней, образующих иерархию. Однако общей справочной модели, подобной модели OSI, для сетей с коммутацией каналов не существует. Например, различные типы телефонных сетей имеют собственные стеки протоколов, отличающиеся количеством уровней и распределением функций между уровнями. Первичные сети, такие как SDH или DWDM, также обладают собственной иерархией протоколов. Ситуация усложняется еще и тем, что практически все типы современных сетей с коммутацией каналов задействуют эту технику только для передачи пользовательских данных, а для управления процессом установления соединений в сети и общего управления сетью применяют технику коммутации пакетов. Такими сетями являются, например, сети ISDN, SDH, DWDM.

Для сетей с коммутацией пакетов сети с коммутацией каналов предоставляют сервис физического уровня, хотя сами они устроены достаточно сложно и поддерживают собственную иерархию протоколов.

Рассмотрим, к примеру, случай, когда несколько локальных пакетных сетей связываются между собой через цифровую телефонную сеть. Очевидно, что функции создания составной сети выполняют протоколы сетевого уровня, поэтому мы устанавливаем в каждой локальной сети маршрутизатор. Маршрутизатор должен быть оснащен интерфейсом, способным установить соединение через телефонную сеть с другой локальной сетью. После того как такое соединение установлено, в телефонной сети образуется поток битов, передаваемых с постоянной скоростью. Это соединение и предоставляет маршрутизаторам сервис физического уровня. Для того чтобы организовать передачу данных, маршрутизаторы используют поверх этого физического канала какой-либо двухточечный протокол канального уровня.

Стандартизация сетей

Универсальный тезис о пользе стандартизации, справедливый для всех отраслей, в компьютерных сетях приобретает особое значение. Суть сети — это соединение разного оборудования, а значит, проблема совместимости является здесь одной из наиболее острых. Без согласования всеми производителями общепринятых стандартов для оборудования и протоколов прогресс в деле «строительства» сетей был бы невозможен. Поэтому все развитие компьютерной отрасли, в конечном счете, отражено в стандартах — любая новая технология только тогда приобретает «законный» статус, когда ее содержание закрепляется в соответствующем стандарте.

В компьютерных сетях идеологической основой стандартизации является рассмотренная ранее модель взаимодействия открытых систем (OSI).

Понятие открытой системы

Что же такое открытая система?
Открытой может быть названа любая система (компьютер, вычислительная сеть, ОС, программный пакет, другие аппаратные и программные продукты), которая построена в соответствии с открытыми спецификациями.

Напомним, что под термином «спецификация» в вычислительной технике понимают формализованное описание аппаратных или программных компонентов, способов их функционирования, взаимодействия с другими компонентами, условий эксплуатации, особых характеристик. Понятно, что не всякая спецификация является стандартом.

Под **открытыми спецификациями** понимаются опубликованные, общедоступные спецификации, соответствующие стандартам и принятые в результате достижения согласия после всестороннего обсуждения всеми заинтересованными сторонами.

Использование при разработке систем открытых спецификаций позволяет третьим сторонам разрабатывать для этих систем различные аппаратные или программные средства расширения и модификации, а также создавать программно-аппаратные комплексы из продуктов разных производителей.

Открытый характер стандартов и спецификаций важен не только для коммуникационных протоколов, но и для разнообразных устройств и программ, выпускаемых для построения сети. Нужно отметить, что большинство стандартов, принимаемых сегодня, носят открытый характер. Время закрытых систем, точные спецификации на которые были известны только фирме-производителю, ушло. Все осознали, что возможность взаимодействия с продуктами конкурентов не снижает, а наоборот, повышает ценность изделия, так как позволяет применять его в большем количестве работающих сетей, собранных из продуктов разных производителей. Поэтому даже такие фирмы, как IBM, Novell и Microsoft, ранее выпускавшие закрытые системы, сегодня активно участвуют в разработке открытых стандартов и применяют их в своих продуктах.

Для реальных систем полная открытость является недостижимым идеалом. Как правило, даже в системах, называемых открытыми, этому определению соответствуют лишь некоторые части, поддерживающие внешние интерфейсы. Например, открытость семейства операционных систем Unix заключается, помимо всего прочего, в наличии стандартизованного программного интерфейса между ядром и приложениями, что позволяет легко переносить приложения из среды одной версии Unix в среду другой версии.

Модель OSI касается только одного аспекта открытости, а именно — открытости средств взаимодействия устройств, связанных в компьютерную сеть. Здесь под открытой системой понимается сетевое устройство, готовое взаимодействовать с другими сетевыми устройствами по стандартным правилам, определяющим формат, содержание и значение принимаемых и отправляемых сообщений.

Если две сети построены с соблюдением принципов открытости, это дает следующие преимущества:

- возможность построения сети из аппаратных и программных средств различных производителей, придерживающихся одного и того же стандарта;
- безболезненная замена отдельных компонентов сети другими, более совершенными, что позволяет сети развиваться с минимальными затратами;
- легкость сопряжения одной сети с другой.

Источники стандартов

Работы по стандартизации вычислительных сетей ведутся большим количеством организаций. В зависимости от статуса организаций различают следующие виды стандартов:

- *стандарты отдельных фирм*, например стек протоколов SNA компании IBM или графический интерфейс OPEN LOOK для Unix-систем компании Sun;
- *стандарты специальных комитетов и объединений* создаются несколькими компаниями, например стандарты технологии ATM, разрабатываемые специально созданным объединением ATM Forum, которое насчитывает около 100 коллективных участников, или стандарты союза Fast Ethernet Alliance, касающиеся технологии 100 Мбит Ethernet;
- *национальные стандарты*, например стандарт FDDI, представляющий один из многочисленных стандартов института ANSI, или стандарты безопасности для операционных систем, разработанные центром NCSC Министерства обороны США;
- *международные стандарты*, например модель и стек коммуникационных протоколов Международной организации по стандартизации (ISO), многочисленные стандарты Международного союза электросвязи (ITU), в том числе стандарты на сети с коммутацией пакетов X.25, сети Frame Relay, ISDN, модемы и многие другие.

Некоторые стандарты, непрерывно развиваясь, могут переходить из одной категории в другую. В частности, фирменные стандарты на продукцию, получившую широкое распространение, обычно становятся международными стандартами де-факто, так как вынуждают производителей из разных стран следовать фирменным стандартам, чтобы обеспечить совместимость своих изделий с этими популярными продуктами. Например, из-за феноменального успеха персонального компьютера компании IBM фирменный стандарт на архитектуру IBM PC стал международным стандартом де-факто.

Более того, ввиду широкого распространения некоторые фирменные стандарты становятся основой для национальных и международных стандартов де-юре. Например, стандарт Ethernet, первоначально разработанный компаниями Digital Equipment, Intel и Xerox, через некоторое время и в несколько измененном виде был принят как национальный стандарт IEEE 802.3, а затем организация ISO утвердила его в качестве международного стандарта ISO 8802.3.

Стандартизация Интернета

Ярким примером открытой системы является Интернет. Эта международная сеть развивалась в полном соответствии с требованиями, предъявляемыми к открытым системам. В разработке ее стандартов принимали участие тысячи специалистов-пользователей этой сети из различных университетов, научных организаций и фирм-производителей вычислительной аппаратуры и программного обеспечения, работающих в разных странах. Само название стандартов, определяющих работу Интернета, — **темы для обсуждения** (Request For Comments, RFC) — показывает гласный и открытый характер принимаемых стандартов. В результате Интернет сумел объединить в себе разнообразное оборудование и программное обеспечение огромного числа сетей, разбросанных по всему миру.

Ввиду постоянной растущей популярности Интернета документы RFC становятся международными стандартами де-факто, многие из которых затем приобретают статус официальных международных стандартов в результате их утверждения какой-либо организацией по стандартизации, как правило, ISO и ITU-T.

Существует несколько организационных подразделений, отвечающих за развитие и, в частности, за стандартизацию архитектуры и протоколов Интернета. Основным из них является научно-административное **сообщество Интернета** (Internet Society, ISOC), объединяющее около 100 000 человек, которое занимается социальными, политическими и техническими проблемами эволюции Интернета.

Под управлением ISOC работает **совет по архитектуре Интернета** (Internet Architecture Board, IAB). В IAB входят две основные группы: Internet Research Task Force (IRTF) и Internet Engineering Task Force (IETF). IRTF координирует долгосрочные исследовательские проекты по протоколам TCP/IP. IETF — это инженерная группа, которая занимается решением текущих технических проблем Интернета. Именно IETF определяет спецификации, которые затем становятся стандартами Интернета. Процесс разработки и принятия стандарта для протокола Интернета состоит из ряда обязательных этапов, или стадий, включающих непременную экспериментальную проверку.

В соответствии с принципом открытости Интернета все документы RFC, в отличие, скажем, от стандартов ISO, находятся в свободном доступе. Список RFC можно найти, в частности, на сайте www.rfc-editor.org.

Стандартные стеки коммуникационных протоколов

Важнейшим направлением стандартизации в области вычислительных сетей является стандартизация коммуникационных протоколов. Наиболее известными стеками протоколов являются: OSI, TCP/IP, IPX/SPX, NetBIOS/SMB, DECnet, SNA (не все из них применяются сегодня на практике).

Стек OSI

Важно различать *модель* OSI и *стек протоколов* OSI. В то время как модель OSI является концептуальной схемой взаимодействия открытых систем, стек OSI представляет собой набор спецификаций конкретных протоколов.

В отличие от других стеков протоколов, стек OSI полностью соответствует модели OSI, включая спецификации протоколов для всех семи уровней взаимодействия, определенных

в этой модели (рис. 4.11). Это и понятно, разработчики стека OSI использовали модель OSI как прямое руководство к действию.

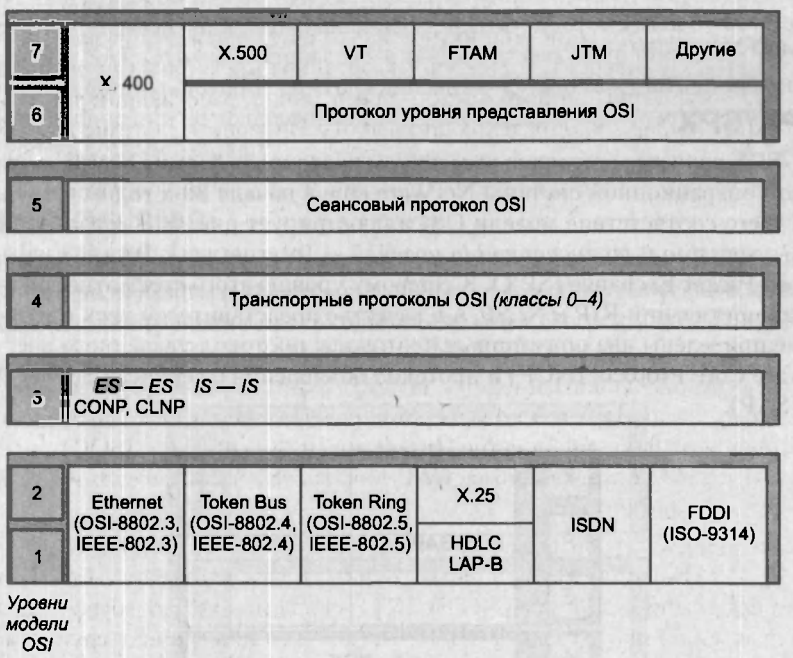


Рис. 4.11. Стек протоколов OSI

Протоколы стека OSI отличает сложность и неоднозначность спецификаций. Эти свойства явились результатом общей политики разработчиков стека, стремившихся учесть в своих протоколах все многообразие уже существующих и появляющихся технологий.

На *физическом* и *канальном уровнях* стек OSI поддерживает протоколы Ethernet, Token Ring, FDDI, а также протоколы LLC, X.25 и ISDN, то есть использует все разработанные вне стека популярные протоколы нижних уровней, как и большинство других стеков.

Сетевой уровень включает сравнительно редко используемые протоколы Connection-oriented Network Protocol (CONP) и Connectionless Network Protocol (CLNP). Как следует из названий, первый из них ориентирован на соединение (connection-oriented), второй — нет (connectionless).

Более популярны протоколы маршрутизации стека OSI: ES-IS (End System — Intermediate System) между конечной и промежуточной системами и IS-IS (Intermediate System — Intermediate System) между промежуточными системами.

Транспортный уровень стека OSI в соответствии с функциями, определенными для него в модели OSI, скрывает различия между сетевыми сервисами с установлением соединения и без установления соединения, так что пользователи получают требуемое качество обслуживания независимо от нижележащего сетевого уровня. Чтобы обеспечить это, транспортный уровень требует, чтобы пользователь задал нужное качество обслуживания.

Службы *прикладного уровня* обеспечивают передачу файлов, эмуляцию терминала, службу каталогов и почту. Из них наиболее популярными являются служба каталогов (стандарт X.500), электронная почта (X.400), протокол виртуального терминала (VTP), протокол передачи, доступа и управления файлами (FTAM), протокол пересылки и управления работами (JTM).

Стек IPX/SPX

Стек IPX/SPX является оригинальным стеком протоколов фирмы Novell, разработанным для сетевой операционной системы NetWare еще в начале 80-х годов. Структура стека IPX/SPX и его соответствие модели OSI иллюстрирует рис. 4.12. Название стеку дали протоколы *сетевого* и *транспортного уровней* – Internetwork Packet Exchange (IPX) и Sequenced Packet Exchange (SPX). К сетевому уровню этого стека отнесены также протоколы маршрутизации RIP и NLSP. А в качестве представителей трех верхних уровней на рисунке приведены два популярных протокола: протокол удаленного доступа к файлам NetWare Core Protocol (NCP) и протокол объявления о сервисах Service Advertising Protocol (SAP).

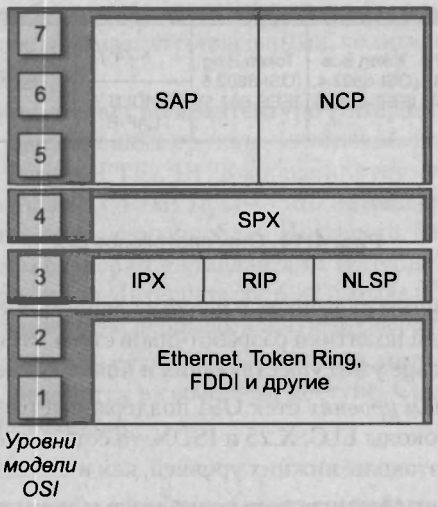


Рис. 4.12. Стек протоколов IPX/SPX

ПРИМЕЧАНИЕ

До 1996 года стек IPX/SPX был бесспорным мировым лидером по числу установленных копий, но затем картина резко изменилась – стек TCP/IP по темпам роста числа установок намного стал опережать другие стеки, а с 1998 года вышел в лидеры и в абсолютном выражении.

Многие особенности стека IPX/SPX обусловлены ориентацией ранних версий ОС NetWare на работу в локальных сетях небольших размеров, состоящих из персональных компьютеров со скромными ресурсами. Понятно, что для таких компьютеров компании Novell нужны были протоколы, на реализацию которых требовалось бы минимальное количество оперативной памяти (ограниченной в IBM-совместимых компьютерах под управлением MS-DOS объемом 640 Кбайт) и которые бы быстро работали на процессорах небольшой

вычислительной мощности. В результате протоколы стека IPX/SPX до недавнего времени отлично справлялись с работой в локальных сетях. Однако в крупных корпоративных сетях они слишком перегружали медленные глобальные связи широковещательными пакетами, интенсивно использующимися несколькими протоколами этого стека, например протоколом SAP. Это обстоятельство, а также тот факт, что стек IPX/SPX является собственностью фирмы Novell и на его реализацию нужно получать лицензию (то есть открытые спецификации не поддерживались), долгое время ограничивали распространенность его только сетями NetWare.

Стек NetBIOS/SMB

Стек NetBIOS/SMB является совместной разработкой компаний IBM и Microsof (рис. 4.13). На физическом и канальном уровнях этого стека также задействованы уже получившие распространение протоколы, такие как Ethernet, Token Ring, FDDI, а на верхних уровнях — специфические протоколы NetBEUI и SMB.

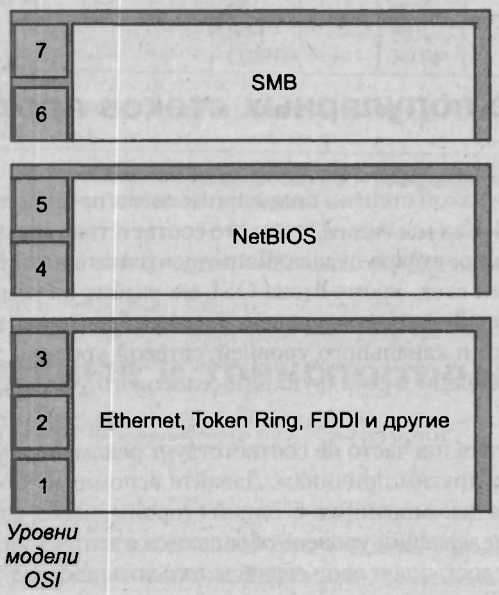


Рис. 4.13. Стек NetBIOS/SMB

Протокол Network Basic Input/Output System (NetBIOS) появился в 1984 году как сетевое расширение стандартных функций базовой системы ввода-вывода (BIOS) IBM PC для сетевой программы PC Network фирмы IBM. В дальнейшем этот протокол был заменен так называемым протоколом расширенного пользовательского интерфейса NetBEUI (NetBIOS Extended User Interface). Для совместимости приложений в качестве интерфейса к протоколу NetBEUI был сохранен интерфейс NetBIOS. NetBEUI разрабатывался как эффективный протокол, потребляющий немного ресурсов и предназначенный для сетей, насчитывающих не более 200 рабочих станций. Этот протокол содержит много полезных сетевых функций, которые можно отнести к транспортному и сеансовому уровням модели OSI, однако с его помощью *невозможна маршрутизация* пакетов. Это ограничивает при-

менение протокола NetBEUI локальными сетями, не разделенными на подсети, и делает невозможным его использование в составных сетях.

Протокол Server Message Block (SMB) поддерживает функции сеансового уровня, уровня представления и прикладного уровня. На основе SMB реализуется файловая служба, а также службы печати и передачи сообщений между приложениями.

Стек TCP/IP

Стек TCP/IP был разработан по инициативе Министерства обороны США более 20 лет назад для связи экспериментальной сети ARPAnet с другими сетями как набор общих протоколов для разнородной вычислительной среды. Большой вклад в развитие стека TCP/IP, который получил свое название по популярным протоколам IP и TCP, внес университет Беркли, реализовав протоколы стека в своей версии ОС Unix. Популярность этой операционной системы привела к широкому распространению протоколов TCP, IP и других протоколов стека. Сегодня этот стек используется для связи компьютеров в Интернете, а также в огромном числе корпоративных сетей. Мы подробно рассмотрим этот стек протоколов в части IV, посвященной сетям TCP/IP.

Соответствие популярных стеков протоколов модели OSI

На рис. 4.14 показано, в какой степени популярные стеки протоколов соответствуют рекомендациям модели OSI. Как мы видим, часто это соответствие весьма условно. В большинстве случаев разработчики стеков отдавали предпочтение скорости работы сети в ущерб модульности — ни один стек, кроме стека OSI, не разбит на семь уровней. Чаще всего в стеке явно выделяются 3–4 уровня: уровень сетевых адаптеров, в котором реализуются протоколы физического и канального уровней, сетевой уровень, транспортный уровень и уровень служб, вбирающий в себя функции сеансового уровня, уровня представления и прикладного уровня.

Структура стеков протоколов часто не соответствует рекомендуемому моделью OSI разбиению на уровни и по другим причинам. Давайте вспомним, чем характеризуется идеальная многоуровневая декомпозиция. С одной стороны, необходимо соблюсти принцип иерархии: каждый вышележащий уровень обращается с запросами только к нижележащему, а нижележащий предоставляет свои сервисы только непосредственно соседствующему с ним вышележащему В стеках протоколов это приводит к тому, что PDU вышележащего уровня всегда инкапсулируется в PDU нижележащего.

С другой же стороны, идеальная многоуровневая декомпозиция предполагает, что все модули, отнесенные к одному уровню, ответственны за решение общей для всех них задачи. Однако эти требования часто вступают в противоречие. Например, основной функцией протоколов сетевого уровня стека TCP/IP (так же как и сетевого уровня OSI) является передача пакетов через составную сеть. Для решения этой задачи в стеке TCP/IP предусмотрено несколько протоколов: протокол продвижения IP-пакетов и протоколы маршрутизации RIP, OSPF и др. Если считать признаком принадлежности к одному и тому же уровню общность решаемых задач, то, очевидно, протокол IP и протоколы маршрутизации должны быть отнесены к одному уровню. Вместе с тем, если принять во внимание, что сообщения протокола RIP инкапсулируются в UDP-дейтаграммы, а сообщения протокола OSPF –

в IP-пакеты, то, следуя формально принципу иерархической организации стека, OSPF следовало бы отнести к транспортному, а RIP — к прикладному уровню. На практике же протоколы маршрутизации обычно включают в сетевой уровень.

<table>
<tr><th>Модель OSI</th><th colspan="2">IBM/Microsoft</th><th colspan="2">TCP/IP</th><th colspan="2">Novell</th><th>Стек OSI</th></tr>
<tr><td>Прикладной</td><td rowspan="2">SMB</td><td rowspan="3">Telnet, FTP, SNMP, SMTP, WWW</td><td rowspan="4">NCP, SAP</td><td>X.400, X.500, FTAM</td></tr>
<tr><td>Представления</td><td>Протокол уровня представления OSI</td></tr>
<tr><td>Сеансовый</td><td rowspan="2">NetBIOS</td><td rowspan="2">TCP</td><td>Сеансовый протокол OSI</td></tr>
<tr><td>Транспортный</td><td>SPX</td><td>Транспортный протокол OSI</td></tr>
<tr><td>Сетевой</td><td colspan="2">IP, RIP, OSPF</td><td>IPX, RIP, NLSP</td><td>ES-ES, IS-IS</td></tr>
<tr><td>Канальный</td><td colspan="5">802.3 (Ethernet), 802.5 (Token Ring), FDDI, ATM, PPP</td></tr>
<tr><td>Физический</td><td colspan="5">Коаксиал, экранированная и неэкранированная витая пара, оптоволокно, радиоволны</td></tr>
</table>

Рис. 4.14. Соответствие популярных стеков протоколов модели OSI

Информационные и транспортные услуги

Услуги компьютерной сети можно разделить на две категории:

- транспортные услуги;
- информационные услуги.

Транспортные услуги состоят в передаче информации между пользователями сети в неизменном виде. При этом сеть принимает информацию от пользователя на одном из своих интерфейсов, передает ее через промежуточные коммутаторы и выдает другому пользователю через другой интерфейс. При оказании транспортных услуг сеть не вносит никаких изменений в передаваемую информацию, передавая ее получателю в том виде, в котором она поступила в сеть от отправителя. Примером транспортной услуги глобальных сетей является объединение локальных сетей клиентов.

Информационные услуги состоят в предоставлении пользователю некоторой новой информации. Информационная услуга всегда связана с операциями по обработке информации: хранению ее в некотором упорядоченном виде (файловая система, база данных), поиску нужной информации и преобразованию информации. Информационные услуги существовали и до появления первых компьютерных сетей, например справочные услуги телефонной сети. С появлением компьютеров информационные услуги пережили революцию, так как компьютер и был изобретен для автоматической программной обработки

информации. Для оказания информационных услуг применяются различные информационные технологии: программирование, управление базами данных и файловыми архивами, веб-сервис, электронная почта.

В телекоммуникационных сетях «докомпьютерной» эры всегда преобладали транспортные услуги. Основной услугой телефонной сети была передача голосового трафика между абонентами, в то время как справочные услуги были дополнительными. В компьютерных сетях одинаково важны обе категории услуг. Эта особенность компьютерных сетей сегодня отражается на названии нового поколения телекоммуникационных сетей, которые появляются в результате конвергенции сетей различных типов. Такие сети все чаще стали называть **инфокоммуникационными**. Это название пока не стало общеупотребительным, но оно хорошо отражает новые тенденции, включая обе составляющие услуг на равных правах.

Деление услуг компьютерных сетей на две категории проявляется во многих аспектах. Существует, например, четкое деление специалистов в области компьютерных сетей на специалистов информационных технологий и сетевых специалистов. К первой категории относятся программисты, разработчики баз данных, администраторы операционных систем, веб-дизайнеры, словом, все, кто имеет дело с разработкой и обслуживанием программного и аппаратного обеспечения компьютеров. Вторая категория специалистов занимается транспортными проблемами сети. Эти специалисты имеют дело с каналами связи и коммуникационным оборудованием, таким как коммутаторы, маршрутизаторы и концентраторы. Они решают проблемы выбора топологии сети, выбора маршрутов для потоков трафика, определения требуемой пропускной способности каналов связи и коммуникационных устройств и другими проблемами, связанными только с передачей трафика через сеть.

Безусловно, каждой категории специалистов необходимо знать проблемы и методы смежной области. Специалисты, занимающиеся разработкой распределенных приложений, должны представлять, какие транспортные услуги они могут получить от сети для организации взаимодействия отдельных частей своих приложений. Например, программист должен понимать, какая из двух предлагаемых стеком TCP/IP транспортных услуг, реализуемых протоколами TCP и UDP, подходит наилучшим образом его приложению. Аналогично, разработчики транспортных средств сети при передаче трафика должны стремиться по максимуму учитывать требования приложений.

Тем не менее специализация в области информационных технологий сохраняется, отражая двойственное назначение компьютерных сетей. Деление услуг сети на транспортные и информационные сказывается и на организации стека протоколов, и на распределении протоколов различных уровней по элементам сети.

Распределение протоколов по элементам сети

На рис. 4.15 показаны основные элементы компьютерной сети: конечные узлы — компьютеры и промежуточные узлы — коммутаторы и маршрутизаторы.

Из рисунка видно, что полный стек протоколов реализован только на конечных узлах, а промежуточные узлы поддерживают протоколы всего трех нижних уровней. Это объясняется тем, что коммуникационным устройствам для продвижения пакетов достаточно функциональности нижних трех уровней. Более того, коммуникационное устройство может поддерживать только протоколы двух нижних уровней или даже одного физического уровня — это зависит от типа устройства.

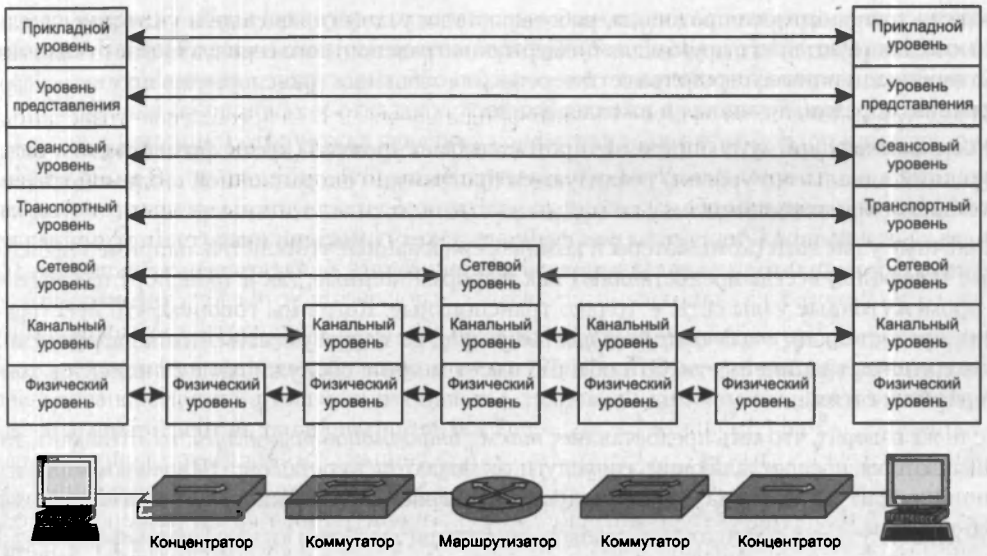


Рис. 4.15. Соответствие функций различных устройств сети уровням модели OSI

Именно к таким устройствам, работающим на физическом уровне, относятся, например, сетевые повторители, называемые также концентраторами, или хабами. Они повторяют электрические сигналы, поступающие на одни их интерфейсы, на других своих интерфейсах, улучшая их характеристики — мощность и форму сигналов, синхронность их следования.

Коммутаторы локальных сетей поддерживают протоколы двух нижних уровней, физического и канального, что дает им возможность работать в пределах стандартных топологий.

Маршрутизаторы должны поддерживать протоколы всех трех уровней, так как сетевой уровень нужен им для объединения сетей различных технологий, а протоколы нижних уровней — для взаимодействия с конкретными сетями, образующими составную сеть, например Ethernet или Frame Relay.

Коммутаторы глобальных сетей (например, ATM), работающие на основе технологии виртуальных каналов, могут поддерживать как два уровня протоколов, так и три. Протокол сетевого уровня нужен им в том случае, если они поддерживают процедуры автоматического установления виртуальных каналов. Так как топология глобальных сетей произвольная, без сетевого протокола обойтись нельзя. Если же виртуальные соединения устанавливаются администраторами сети вручную, то коммутатору глобальной сети достаточно поддерживать только протоколы физического и канального уровней, чтобы передавать данные по уже проложенным виртуальным каналам.

Компьютеры, на которых работают сетевые приложения, должны поддерживать протоколы всех уровней. Протоколы прикладного уровня, пользуясь сервисами протоколов уровня представления и сеансового уровня, предоставляют приложениям набор сетевых услуг в виде сетевого прикладного программного интерфейса (API). Протокол транспортного уровня также работает на всех конечных узлах. При передаче данных через сеть два

модуля транспортного протокола, работающие на узле-отправителе и узле-получателе, взаимодействуют друг с другом для поддержания транспортного сервиса нужного качества. Коммуникационные устройства сети переносят сообщения транспортного протокола прозрачным образом, не вникая в их содержание.

В компьютерах коммуникационные протоколы всех уровней (кроме физического и части функций канального уровня) реализуются программно операционной системой или системными приложениями.

Конечные узлы сети (компьютеры и компьютеризованные устройства, например мобильные телефоны) всегда предоставляют как информационные, так и транспортные услуги, а промежуточные узлы сети — только транспортные. Когда мы говорим, что некоторая сеть предоставляет *только транспортные услуги*, то подразумеваем, что конечные узлы находятся за границей сети. Это обычно имеет место в обслуживающих клиентов коммерческих сетях.

Если же говорят, что сеть предоставляет *также информационные услуги*, то это значит, что компьютеры, предоставляющие эти услуги, включаются в состав сети. Примером является типичная ситуация, когда поставщик услуг Интернета поддерживает еще и собственные веб-серверы.

Вспомогательные протоколы транспортной системы

Настало время сказать, что на рис. 4.15 показан упрощенный вариант распределения протоколов между элементами сети. В реальных сетях некоторые из коммуникационных устройств поддерживают не только протоколы трех нижних уровней, но и протоколы верхних уровней. Так, маршрутизаторы реализуют протоколы маршрутизации, позволяющие автоматически строить таблицы маршрутизации, а концентраторы и коммутаторы часто поддерживают протоколы SNMP и telnet, которые не нужны для выполнения основных функций этих устройств, но позволяют конфигурировать их и управлять ими удаленно. Все эти протоколы являются протоколами прикладного уровня и выполняют некоторые вспомогательные (служебные) функции транспортной системы. Очевидно, что для работы прикладных протоколов сетевые устройства должны также поддерживать протоколы промежуточных уровней, таких как IP и TCP/UDP.

Вспомогательные протоколы можно разделить на группы в соответствии с их функциями.

- Первую группу вспомогательных протоколов представляют протоколы *маршрутизации*, такие как RIP, OSPF, BGP. Без этих протоколов маршрутизаторы не смогут продвигать пакеты, так как таблица маршрутизации будет пустой (если только администратор не заполнит ее вручную, но это не очень хорошее решение для крупной сети). Если рассматривать не только стек TCP/IP, но и стеки протоколов сетей с виртуальными каналами, то в эту группу попадают служебные протоколы, которые используются для установления виртуальных каналов.
- Другая группа вспомогательных протоколов выполняет *преобразование адресов*. Здесь работает протокол DNS, который преобразует символьные имена узлов в IP-адреса. Протокол DHCP позволяет назначать IP-адреса узлам динамически, а не статически, что облегчает работу администратора сети.

- Третью группу образуют протоколы, которые используются для *управления сетью*. В стеке TCP/IP здесь находится протокол SNMP (Simple Network Management Protocol – простой протокол управления сетью), который позволяет автоматически собирать информацию об ошибках и отказах устройств, а также протокол Telnet, с помощью которого администратор может удаленно конфигурировать коммутатор или маршрутизатор.

При рассмотрении вспомогательных протоколов мы столкнулись с ситуацией, когда деления протоколов на уровни иерархии (то есть деления «по вертикали»), присущего модели OSI, оказывается недостаточно. Полезным оказывается деление протоколов на группы «по горизонтали».

И хотя такое деление отсутствует в модели OSI, оно существует в других стеках протоколов. Например, при стандартизации сетей ISDN, которые, как мы уже упоминали, используют как принцип коммутации пакетов, так и принцип коммутации каналов, все протоколы разделяют на три слоя (рис. 4.16):

- **пользовательский слой** (user plane) образует группа протоколов, предназначенных для того, чтобы переносить пользовательский голосовой трафик;
- **слой управления** (control plane) составляют протоколы, необходимые для установления соединений в сети;
- **слой менеджмента** (management plane) объединяет протоколы, поддерживающие операции менеджмента, такие как анализ ошибок и конфигурирование устройств.

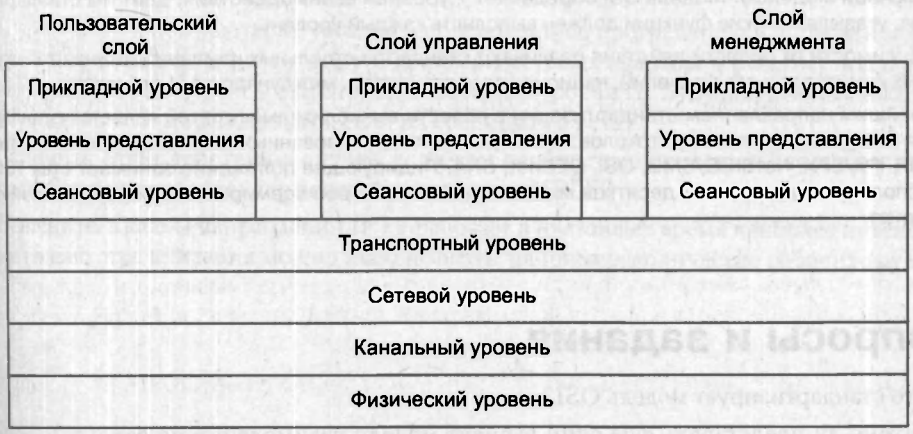


Рис. 4.16. Три группы протоколов

И хотя такое «горизонтальное» деление протоколов пока не является общепринятым для компьютерных сетей, оно полезно, так как позволяет глубже понять назначение протоколов. Кроме того, оно объясняет сложности, возникающие при соотнесении некоторых протоколов уровням модели OSI. Например, в книгах одних авторов протоколы маршрутизации могут находиться на сетевом уровне, в книгах других — на прикладном. Это происходит не из-за небрежности авторов, а из-за объективных трудностей классификации. Модель OSI хорошо подходит для стандартизации протоколов, которые переносят пользовательский трафик, то есть протоколов пользовательского слоя. В то же время она в гораздо меньшей степени годится для стандартизации вспомогательных протоколов.

Поэтому многие авторы и помещают протоколы маршрутизации на сетевой уровень, чтобы каким-то образом отразить функциональную близость этих протоколов к транспортным услугам сети, которые реализуются протоколом IP.

Выводы

Эффективной моделью средств взаимодействия компьютеров в сети является многоуровневая структура, в которой модули вышележащего уровня при решении своих задач рассматривают средства нижележащего уровня как некий инструмент. Каждый уровень данной структуры поддерживает интерфейсы двух типов. Во-первых, это интерфейсы услуг с выше- и нижележащими уровнями «своей» иерархии средств. Во-вторых, это одноранговый интерфейс со средствами другой взаимодействующей стороны, расположенными на том же уровне иерархии. Этот интерфейс называют протоколом.

Иерархически организованный набор протоколов, достаточный для взаимодействия узлов в сети, называется стеком протоколов. Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней — как правило, программными средствами. Программный модуль, реализующий некоторый протокол, называют протокольной сущностью, или тоже протоколом.

В начале 80-х годов ISO, ITU-T при участии некоторых других международных организаций по стандартизации разработали стандартную модель взаимодействия открытых систем (OSI). Модель OSI содержит описание обобщенного представления средств сетевого взаимодействия и используется в качестве своего рода универсального языка сетевых специалистов, именно поэтому ее называют справочной моделью. Модель OSI определяет 7 уровней взаимодействия, дает им стандартные имена, указывает, какие функции должен выполнять каждый уровень.

В зависимости от области действия различают стандарты отдельных компаний, стандарты специальных комитетов и объединений, национальные стандарты, международные стандарты.

Важнейшим направлением стандартизации в области вычислительных сетей является стандартизация коммуникационных протоколов. Примерами стандартизованных стеков протоколов являются TCP/IP, IPX/SPX, NetBIOS/SMB, OSI, DECnet, SNA. Лидирующее положение занимает стек TCP/IP, он используется для связи десятков миллионов компьютеров всемирной информационной сети Интернет.

Вопросы и задания

1. Что стандартизирует модель OSI?
2. Можно ли представить еще один вариант модели взаимодействия открытых систем с другим количеством уровней, например 8 или 5?
3. Какие из приведенных утверждений не всегда справедливы:
   а) протокол – это стандарт, описывающий правила взаимодействия двух систем;
   б) протокол — это формализованное описание правил взаимодействия, включая последовательность обмена сообщениями и их форматы;
   в) логический интерфейс — это формализованное описание правил взаимодействия, включая последовательность обмена сообщениями и их форматы.
4. На каком уровне модели OSI работает прикладная программа?

5. Как вы считаете, протоколы транспортного уровня устанавливаются:

   а) только на конечных узлах;

   б) только на промежуточном коммуникационном оборудовании (маршрутизаторах);

   в) и там, и там.

6. На каком уровне модели OSI работают сетевые службы?
7. Назовите известные вам организации, работающие в области стандартизации компьютерных сетей.
8. Какие из перечисленных терминов являются синонимами:

   а) стандарт;
   б) спецификация;
   в) RFC;
   г) все;
   д) никакие.

9. К какому типу стандартов могут относиться современные документы RFC? Варианты ответов:

   а) к стандартам отдельных фирм;

   б) к государственным стандартам;

   в) к национальным стандартам;

   г) к международным стандартам.

10. Пусть на двух компьютерах установлено идентичное программное и аппаратное обеспечение за исключением того, что драйверы сетевых адаптеров Ethernet поддерживают разные интерфейсы с протоколом сетевого уровня IP. Будут ли эти компьютеры нормально взаимодействовать, если их соединить в сеть?
11. Опишите ваши действия в случае, если вам необходимо проверить, на каком этапе находится процесс стандартизации технологии MPLS?
12. Выясните, в каком направлении IETF работает в настоящее время наиболее интенсивно (в качестве критерия можно использовать, например, количество рабочих групп)?

ГЛАВА 5 Примеры сетей

В этой главе рассматриваются примеры наиболее популярных типов сетей — сетей операторов связи, корпоративных сетей и Интернет.

Глава начинается с классификации типов сетей, которая дает общее представление о наиболее существенных признаках, характеризующих сети.

Несмотря на различия между этими типами сетей, они имеют много общего и, прежде всего, схожую структуру. Поэтому перед обсуждением каждого из перечисленных типов сетей мы рассмотрим обобщенную структуру телекоммуникационной сети.

Заканчивается глава описанием Интернета. Эта сеть, уникальная во многих отношениях, оказала критическое влияние на развитие сетевых технологий в современном мире.

Классификация компьютерных сетей

Классификация — процесс группирования (отнесения к тому или иному типу) объектов изучения в соответствии с их общими признаками.

Каждый реальный объект может быть наделен множеством признаков. *Субъективный* характер любой классификации проявляется в том, что имеется некоторый произвол при выборе среди этого множества признаков тех, которые будут использованы для классификации, то есть при выборе **критериев классификации**. Приведенная далее классификация компьютерных сетей не является исключением — в других книгах вы можете встретить другие классификации, основанные на критериях, отличающихся от выбранных здесь.

Классификация компьютерных сетей в технологическом аспекте

Итак, начнем с самой разветвленной классификации компьютерных сетей, в которой в качестве критериев классификации используются различные технологические характеристики сетей, такие как топология, метод коммутации, метод продвижения пакетов, тип среды передачи и др. Такая классификация может быть названа классификацией в *технологическом аспекте*.

- Поскольку до недавнего времени выбор технологии, используемой для построения сети, был в первую очередь обусловлен ее территориальным масштабом, мы начнем нашу классификацию с технологических признаков компьютерной сети, обусловленных *территорией покрытия*. Все сети по этому критерию можно разделить на две группы:
  - **локальные сети** (Local Area Network, LAN);
  - **глобальные сети** (Wide Area Network, WAN).

  Первые локальные и глобальные сети представляли собой два существенно отличающихся технологических направления. Мы уже обозначили особенности двух этих направлений, когда рассматривали эволюцию компьютерных сетей (см. главу 1). В частности, в локальных сетях обычно используются более качественные линии связи, которые не всегда доступны (из-за экономических ограничений) на больших расстояниях, свойственных глобальным сетям. Высокое качество линий связи в локальных сетях позволило упростить процедуры передачи данных за счет применения немодулированных сигналов и отказа от обязательного подтверждения получения пакета. Благодаря этому скорость обмена данными между конечными узлами в локальных сетях, как правило, выше, чем в глобальных. Несмотря на то что процесс сближения технологий локальных и глобальных сетей идет уже давно, различия между этими технологиями все еще достаточно отчетливы, что и дает основания относить соответствующие сети к различным технологическим типам.

  Подчеркнем, что говоря в данном контексте «локальные сети» или «глобальные сети», мы имеем в виду, прежде всего, *различия технологий* локальных и глобальных сетей, а не тот факт, что эти сети имеют разный территориальный масштаб.

  Мы также уже упоминали ранее о так называемых **городских сетях**, или **сетях мегаполиса** (Metropolitan Area Network, MAN). Эти сети предназначены для обслуживания

территории крупного города — мегаполиса, и сочетают в себе признаки как локальных, так и глобальных сетей. От первых они унаследовали большую плотность подключения конечных абонентов и высокоростные линии связи, а от последних — большую протяженность линий связи. В то же время появление городские сетей не привело к возникновению каких-нибудь качественно новых технологий[1], поэтому мы не выделили их в отдельный технологический тип сетей.

- В соответствии с технологическими признаками, обусловленными *средой передачи*, компьютерные сети подразделяют на два класса:
  - **проводные сети**, то есть сети, каналы связи которых построены с использованием медных или оптических кабелей;
  - **беспроводные сети**, то есть сети, в которых для связи используются беспроводные каналы связи, например радио, СВЧ, инфракрасные или лазерные каналы.

  Тип среды передачи влияет на технологию компьютерной сети, так как ее протоколы должны учитывать скорость и надежность соединения, обеспечиваемого каналом, а также частоту искажения в нем битов информации. Как вы уже знаете, различие технологий локальных и глобальных сетей во многом определялось различием качества используемых в этих сетях каналов связи.

  Качество канала связи зависит от многих факторов, но наиболее кардинально на него влияет выбор проводной или беспроводной среды.

  Любая беспроводная среда — будь то радиоволны, инфракрасные лучи или СВЧ-сигналы спутниковой связи — гораздо больше подвержена влиянию внешних помех, чем проводная. Роса, туман, солнечные бури, работающие в комнате микроволновые печи — вот только несколько примеров источников помех, которые могут привести к резкому ухудшению качества беспроводного канала. А значит, технологии беспроводных сетей должны учитывать типичность таких ситуаций и строиться таким образом, чтобы обеспечивать работоспособность сети, несмотря на ухудшение вешних условий. Кроме того, существует ряд других специфических особенностей беспроводных сетей, которые служат основанием для выделения их в особый класс, например естественное разделение радиосреды узлами сети, находящимися в радиусе действия всенаправленного передатчика; распределение диапазона радиочастот между сетями различного назначения, например между телефонными и компьютерными.

- В зависимости от способа *коммутации* сети подразделяются на два класса:
  - **сети с коммутацией пакетов**;
  - **сети с коммутацией каналов**.

  Вы уже знакомы с особенностями и отличиями методов коммутации пакетов и каналов, поэтому не удивительно, что эти методы приводят к существованию двух фундамен-

---

[1] Существует новая технология, которая первоначально была задумана как технология городских сетей и поэтому в течение некоторого времени была известна как Metro Ethernet. Однако со временем ее назначение было расширено (или, если угодно, — трансформировано), и теперь она продолжает разрабатываться под названием Carrier Grade Ethernet (см. главу 21), то есть Ethernet операторского класса. Такое изменение названия хорошо иллюстрирует относительность классификации технологий — по сути, это та же самая технология, но названная в соответствии с другим ее признаком, который оказался более существенным.

тально различных типов сетей: хотя в компьютерных сетях преимущественно используется техника коммутации пакетов, принципиально допустимо и применение в них техники коммутации каналов.

В свою очередь, техника коммутации пакетов допускает несколько вариаций, отличающихся способом продвижения пакетов:

- **дейтаграммные сети**, например Ethernet;
- **сети, основанные на логических соединениях**, например IP-сети, использующие на транспортном уровне протокол TCP;
- **сети, основанные на виртуальных каналах**, например MPLS-сети.

- Сети могут быть классифицированы на основе *топологии*. Топологический тип сети весьма отчетливо характеризует сеть, он понятен как профессионалам, так и пользователям. Мы подробно рассматривали базовые топологии сетей, поэтому здесь только перечислим их: **полносвязная топология**, **дерево**, **звезда**; **кольцо**, **смешанная топология**.
- Компьютерные сети разделяют также по признаку их *первичности*:
  - **Первичные сети** занимают особое положение в мире телекоммуникационных сетей, это своего рода *вспомогательные* сети, которые нужны для того, чтобы гибко создавать постоянные физические двухточечные каналы для других компьютерных и телефонных сетей. В соответствии с семиуровневой моделью OSI первичные сети подобно простым кабелям выполняют функции физического уровня сетей. Однако в отличие от кабелей первичные сети включают дополнительное коммуникационное оборудование, которое путем соответствующего конфигурирования позволяет прокладывать новые физические каналы между конечными точками сети. Другими словами, первичная сеть — это гибкая среда для создания физических каналов связи.
  - **Наложенные сети** в этой классификации — это все остальные сети, которые предоставляют услуги конечным пользователям и строятся на основе каналов первичных сетей — «накладываются» поверх этих сетей. То есть и компьютерные, и телефонные, и телевизионные сети являются наложенными.

Другие аспекты классификации компьютерных сетей

Сети можно классифицировать в зависимости от того, кому предназначаются услуги этих сетей. Впервые мы имеем дело не с техническим, а организационным критерием классификации.

- Итак, в зависимости от того, *какому типу пользователей предназначаются услуги сети*, сети делятся на два класса: сети операторов связи и корпоративные сети.
  - **Сети операторов связи** предоставляют публичные услуги, то есть клиентом сети может стать любой индивидуальный пользователь или любая организация, которая заключила соответствующий коммерческий договор на предоставление той или иной телекоммуникационной услуги. Традиционными услугами операторов связи являются услуги телефонии, а также предоставления каналов связи в аренду тем организациям, которые собираются строить на их основе собственные сети. С рас-

пространением компьютерных сетей операторы связи существенно расширили спектр своих услуг, добавив доступ в Интернет, услуги виртуальных частных сетей, веб-хостинг, электронную почту и IP-телефонию, а также широковещательную рассылку аудио- и видеосигналов. Ввиду того, что сеть оператора связи обслуживает, как правило, больше клиентов, чем корпоративная сеть (бывают, конечно, и исключения из этого правила), и, кроме того, оператор несет прямую материальную ответственность за сбои в работе своей сети, существует неформальное понятие «оборудование операторского класса», отражающее высокие показатели надежности, управляемости и производительности такого оборудования.

- **Корпоративные сети** предоставляют услуги только сотрудникам предприятия, которое владеет этой сетью. Хотя формально корпоративная сеть может иметь любой размер, обычно под корпоративной понимают сеть крупного предприятия, которая состоит как из локальных сетей, так и из объединяющей их глобальной сети.

- В зависимости от *функциональной роли в составной сети* сети делятся на три класса: сети доступа, магистральные сети и сети агрегирования трафика.
  - **Сети доступа** — это сети, предоставляющие доступ индивидуальным и корпоративным абонентам от их помещений (квартир, офисов) до первого помещения (пункта присутствия) оператора сети связи или оператора корпоративной сети. Другими словами, это сети, ответственные за расширение глобальной сети до помещений ее клиентов.
  - **Магистральные сети** — это сети, представляющие собой наиболее скоростную часть (ядро) глобальной сети, которая объединяет многочисленные сети доступа в единую сеть.
  - **Сети агрегирования трафика** — это сети, агрегирующие данные от многочисленных сетей доступа для компактной передачи их по небольшому числу каналов связи в магистраль. Сети агрегирования обычно используются только в крупных глобальных сетях, где они занимают промежуточную позицию, помогая магистральной сети обрабатывать трафик, поступающий от большого числа сетей доступа. В сетях среднего и небольшого размера сети агрегирования обычно отсутствуют.

Заканчивая обсуждение классификации компьютерных сетей, мы еще раз подчеркиваем *условный характер* любой классификации. Даже в этой книге вы еще не раз столкнетесь с другими, также достаточно широко распространенными критериями классификации. Заметим также, что в общем случае критерии классификации *не зависят друг от друга*. А это означает, что если согласно какому-то из критериев две сети принадлежат к одному и тому же типу, то при классификации по другому критерию они могут быть отнесены к разным типам. Рассмотрим, например, предприятие, у которого имеется много небольших филиалов в разных городах. Сеть каждого филиала располагается в пределах одного здания и по критерию территориального покрытия относится к классу локальных сетей. Данная организация обладает также сетью, которая связывает все локальные сети филиалов в единую сеть, покрывающую большую территорию, и по данному признаку относится к классу глобальных сетей. В то же время все сети рассматриваемой организации (и сети филиалов и связывающая их сеть) входят в один и тот же класс — класс корпоративных сетей.

Обобщенная структура телекоммуникационной сети

Несмотря на сохраняющиеся различия между компьютерными, телефонными, телевизионными, радио и первичными сетями, в их структуре можно найти много общего. В общем случае телекоммуникационная сеть состоит из следующих компонентов (рис. 5.1):

- терминального оборудования пользователей (возможно, объединенного в сеть);
- сетей доступа;
- магистральной сети;
- информационных центров, или центров управления сервисами (Services Control Point, SCP).

Рис. 5.1. Обобщенная структура телекоммуникационной сети

Сеть доступа

Сеть доступа составляет нижний уровень иерархии телекоммуникационной сети. Основное назначение сети доступа — *концентрация информационных потоков*, поступающих по многочисленным каналам связи от оборудования клиентов, в сравнительно небольшом количестве узлов магистральной сети.

В случае компьютерной сети **терминальным оборудованием** являются компьютеры, телефонной — телефонные аппараты, телевизионной или радиосети — соответствующие теле- или радиоприемники. Терминальное оборудование пользователей может быть объединено в сети, которые не включаются в состав телекоммуникационной сети, так как принадлежат пользователям и размещаются на их территории. Компьютеры пользователей объединяются в LAN, а телефоны могут подключаться к **офисному телефонному коммутатору** (Private Branch Exchange, PBX).

Сеть доступа — это региональная сеть, отличающаяся большой разветвленностью. Как и телекоммуникационная сеть в целом, сеть доступа может состоять из нескольких уровней (на рис. 5.1 их показано два). Коммутаторы[1], установленные в узлах нижнего уровня, мультиплексируют информацию, поступающую по многочисленным абонентским каналам, часто называемым **абонентскими окончаниями**, и передают ее коммутаторам верхнего уровня, чтобы те, в свою очередь, передали ее коммутаторам магистрали.

Количество уровней сети доступа зависит от ее размера; небольшая сеть доступа может состоять из одного уровня, крупная — из двух-трех.

Магистральная сеть

Магистральная сеть объединяет отдельные сети доступа, обеспечивая транзит трафика между ними по высокоскоростным каналам.

Коммутаторы магистрали могут оперировать не только информационными соединениями между отдельными пользователями, но и агрегированными информационными потоками, переносящими данные большого количества пользовательских соединений. В результате информация с помощью магистрали попадает в сеть доступа получателей, где она демультиплексируется и коммутируется таким образом, чтобы на входной порт оборудования пользователя поступала только адресованная ему информация.

ПРИМЕР-АНАЛОГИЯ

Вы можете легко заметить, что любая национальная сеть автомобильных дорог имеет ту же иерархическую структуру, что и крупная телекоммуникационная сеть. Обычно существует разветвленная инфраструктура небольших дорог, связывающих деревни и поселки. Эти дороги довольно узкие, так как интенсивность трафика между этими населенными пунктами невысока и нет смысла делать подобные дороги многорядными. Такие дороги вливаются в более скоростные и соответственно более широкие дороги, которые, в свою очередь, имеют соединения с национальными супермагистралями. Иерархия автомобильных дорог, как и иерархия телекоммуникационных сетей, отражает интенсивность трафика между отдельными населенными пунктами и регионами страны и делает автомобильное движение более эффективным.

Информационные центры

Информационные центры, или **центры управления сервисами**, реализуют информационные услуги сети. В таких центрах может храниться информация двух типов:

- пользовательская информация, то есть информация, которая непосредственно интересует конечных пользователей сети;

[1] Термин «коммутатор» используется здесь в широком смысле.

- вспомогательная служебная информация, помогающая поставщику услуг предоставлять услуги пользователям.

Примером информационных ресурсов первого типа могут служить веб-порталы, на которых расположена разнообразная справочная и новостная информация, информация электронных магазинов и т. п. В телефонных сетях подобные центры оказывают услуги экстренного вызова (например, милиции, скорой помощи) и справочные услуги различных организаций и предприятий — вокзалов, аэропортов, магазинов и т. п.

К информационным центрам, хранящим ресурсы второго типа, можно отнести, например, различные системы аутентификации и авторизации пользователей, с помощью которых организация, владеющая сетью, проверяет права пользователей на получение тех или иных услуг; системы биллинга, подсчитывающие в коммерческих сетях плату за полученные услуги; базы данных учетной информации пользователей, хранящие имена и пароли, а также перечни услуг, на которые подписан каждый пользователь. В телефонных сетях существуют централизованные центры управления сервисами (SCP), в которых установлены компьютеры, хранящие программы нестандартной обработки телефонных вызовов пользователей, например вызовов к бесплатным справочным услугам коммерческих предприятий (так называемые услуги 800) или вызовов при проведении телеголосования.

Естественно, у сетей каждого конкретного типа имеется много особенностей, тем не менее их структура в целом соответствует описанной. В то же время, в зависимости от назначения и размера сети, в ней могут отсутствовать или же иметь несущественное значение некоторые составляющие обобщенной структуры. Например, в небольшой локальной компьютерной сети нет ярко выраженных сетей доступа и магистрали — они сливаются в общую и достаточно простую структуру. В корпоративной сети, как правило, отсутствует система биллинга, так как услуги сотрудником предприятия оказываются не на коммерческой основе. В некоторых телефонных сетях могут отсутствовать информационные центры, а в телевизионных сетях сеть доступа приобретает вид распределительной сети, так как информация в ней распространяется только в одном направлении — из сети к абонентам.

Сети операторов связи

Как уже отмечалось, важным признаком классификации сетей является получатель услуг, предоставляемых сетью. **Сети операторов связи** (поставщиков услуг) оказывают общедоступные услуги, а **корпоративные сети** — услуги сотрудникам только того предприятия, которое владеет сетью.

Специализированное предприятие, которое создает телекоммуникационную сеть для оказания общедоступных услуг, владеет этой сетью и поддерживает ее работу, называется **оператором связи** (telecommunication carrier).

Операторы связи осуществляют свою деятельность на коммерческой основе, заключая договоры с потребителями услуг.

Операторы связи отличаются друг от друга:

- набором предоставляемых услуг;
- территорией, в пределах которой предоставляются услуги;

- типом клиентов, на которых ориентированы их услуги;
- имеющейся во владении оператора инфраструктурой — линиями связи, коммутационным оборудованием, информационными серверами и т. п.;
- отношением к монополии на предоставление услуг.

Услуги

Особенностью современных операторов связи является то, что они, как правило, оказывают услуги нескольких типов, например услуги телефонии и доступа в Интернет. Услуги можно разделить на несколько уровней и групп. На рис. 5.2 показаны только некоторые основные уровни и группы, но и эта неполная картина хорошо иллюстрирует широту спектра современных телекоммуникационных услуг и сложность их взаимосвязей. Услуги более высокого уровня опираются на услуги нижележащих уровней. Группы услуг выделены по типу сетей, которые их оказывают, — телефонные или компьютерные (для полноты картины нужно было бы дополнить рисунок услугами телевизионных и радио сетей).

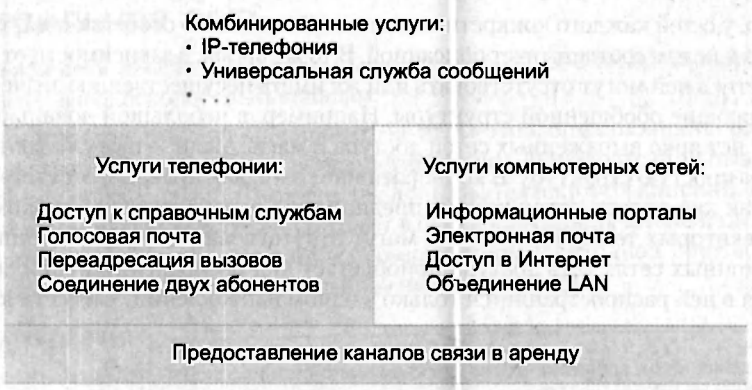


Рис. 5.2. Классификация услуг телекоммуникационной сети (закрашенные области соответствуют традиционным услугам операторов связи)

Услуги предоставления каналов связи в аренду являются услугами самого нижнего уровня, так как заставляют пользователя дополнительно строить с помощью предоставленных каналов собственную сетевую инфраструктуру (установить телефонные коммутаторы или коммутаторы пакетных сетей), прежде чем начать извлекать из них какую-либо выгоду. Обычно такими услугами пользуются либо другие операторы связи, не имеющие собственных каналов связи, либо крупные корпорации, которые на базе каналов строят свои частные корпоративные сети (см. далее).

Следующий уровень составляют две большие группы услуг: телефонные услуги и услуги компьютерных сетей.

Телефонные услуги и предоставление каналов связи в аренду на протяжении очень долгого времени были традиционным набором услуг оператора связи.

Услуги компьютерных сетей стали предлагаться намного позже, чем телефонные, и по абсолютному уровню доходов, приносимых операторам связи, они пока значительно отстают

от традиционных телефонных услуг. Тем не менее подавляющее большинство операторов связи предоставляет услуги компьютерных сетей, и по темпам роста они намного опережают традиционные телефонные услуги, имея отличные перспективы. В объемном исчислении всемирный трафик компьютерных данных уже превзошел телефонный трафик, но низкие тарифы на услуги передачи данных пока не позволяют им догнать традиционные услуги в стоимостном выражении.

Каждый из описанных уровней услуг, в свою очередь, можно разделить на подуровни. Например, оператор может предоставлять предприятию-клиенту на основе услуги доступа в Интернет, которая заключается в простом подключении компьютера или локальной сети ко всемирной общедоступной сети, такие дополнительные услуги, как организацию виртуальной частной сети, надежно защищенной от остальных пользователей Интернета, или же создание информационного веб-портала предприятия и размещение его в своей сети.

Верхний уровень сегодня занимают комбинированные услуги, реализация которых требует совместного оперативного взаимодействия компьютерных и телефонных сетей. Ярким примером таких услуг является международная IP-телефония, которая отобрала у традиционной международной телефонии значительную часть клиентов.

Комбинированные услуги — это прямое следствие конвергенции сетей и главная движущая сила этого процесса.

Услуги можно разделить и по другому принципу — на **транспортные** и **информационные**. Телефонный разговор — это пример услуги первого типа, так как оператор доставляет голосовой трафик от одного абонента к другому. Примерами информационных услуг являются справочные услуги телефонной сети или веб-сайтов.

Именно этот тип различий услуг отражается в названиях телекоммуникационных компаний. Мы говорим «оператор» применительно к традиционным компаниям, основным бизнесом которых всегда были телефонные услуги и услуги предоставления каналов связи в аренду, то есть транспортные услуги. Название «поставщик услуг», или «провайдер», стало популярным с массовым распространением Интернета и его информационной услуги WWW.

Услуги можно различать не только по виду предоставляемой информации, но и по степени их интерактивности. Так, телефонные сети оказывают **интерактивные услуги**, поскольку два абонента, участвующие в разговоре (или несколько абонентов, если это конференция), попеременно проявляют активность. Аналогичные услуги предоставляют компьютерные сети, пользователи которых могут активно участвовать в просмотре содержания веб-сайта, отвечая на вопросы анкеты или играя в игры.

В то же время радиосети и телевизионные сети оказывают **широковещательные услуги**, при этом информация распространяется только в одну сторону — из сети к абонентам по схеме «один ко многим».

Клиенты

Все множество клиентов — потребителей инфотелекоммуникационных услуг — можно разделить на два больших лагеря: массовые **индивидуальные клиенты** и **корпоративные клиенты**.

В первом случае местом потребления услуг выступает квартира или частный дом, а клиентами — жильцы, которым нужны, прежде всего, базовые услуги — телефонная связь, теле-

видение, радио, доступ в Интернет. Для **массовых клиентов** очень важна экономичность услуги — низкая месячная оплата, возможность использования стандартных терминальных устройств, таких как телефонные аппараты, телевизионные приемники, персональные компьютеры, а также возможность использования существующей в квартире проводки в виде телефонной пары и телевизионного коаксиального кабеля. Сложные в обращении и дорогие терминальные устройства, такие как, например, компьютеризованные телевизоры или IP-телефоны, вряд ли станут массовыми до тех пор, пока не приблизятся по стоимости к обычным телевизорам или телефонам и не будут иметь простой пользовательский интерфейс, не требующий для его освоения прослушивания специальных курсов. Существующая в наших домах проводка — это серьезное ограничение для предоставления услуг доступа в Интернет и новых услуг компьютерных сетей, так как она не была рассчитана на передачу данных, а подведение к каждому дому нового качественного кабеля, например волоконно-оптического, — дело дорогое. Поэтому доступ в Интернет чаще всего предоставляется через существующие в доме окончания телефонной сети: либо с помощью низкоскоростного аналогового модемного соединения через телефонную сеть «из конца в конец», либо с помощью новых более скоростных цифровых технологий доступа, в которых используется телефонная сеть, а именно — абонентское окончание телефонной линии, но только на первом этапе, а далее данные передаются в обход телефонной сети по компьютерной сети с коммутацией пакетов. Существуют также технологии доступа, в которых для передачи данных применяется имеющаяся в городе сеть кабельного телевидения.

Корпоративные клиенты — это предприятия и организации различного профиля. Мелкие предприятия по набору требуемых услуг не слишком отличаются от массовых клиентов — это те же базовые телефония и телевидение, стандартный модемный доступ к информационным ресурсам Интернета. Разве что телефонных номеров такому предприятию может потребоваться не один, а два или три.

Крупные же предприятия, состоящие из нескольких территориально рассредоточенных отделений и филиалов, а также имеющие сотрудников, часто работающих дома, нуждаются в расширенном наборе услуг. Прежде всего, такой услугой является **виртуальная частная сеть** (Virtual Private Network, VPN), в которой оператор связи создает для предприятия иллюзию того, что все его отделения и филиалы соединены частной сетью, то есть сетью, полностью принадлежащей предприятию-клиенту и полностью управляемой предприятием-клиентом. На самом же деле для этих целей используется сеть оператора, то есть общедоступная сеть, которая одновременно передает данные многих клиентов.

В последнее время корпоративные пользователи все чаще получают не только транспортные, но и информационные услуги операторов, например переносят собственные веб-сайты и базы данных на территорию оператора, поручая последнему поддерживать их работу и обеспечивать быстрый доступ к ним для сотрудников предприятия и, возможно, других пользователей сети оператора.

Инфраструктура

Помимо субъективных причин на формирование набора предлагаемых оператором услуг оказывает серьезное влияние материально-технический фактор. Так, для оказания услуг по аренде каналов оператор должен иметь в своем распоряжении транспортную сеть, например первичную сеть PDH/SDH, а для оказания информационных веб-услуг — собственный сайт в Интернете.

В тех случаях, когда у оператора отсутствует вся необходимая инфраструктура для оказания некоторой услуги, он может воспользоваться возможностями другого оператора; на базе этих возможностей, а также собственных элементов инфраструктуры требуемая услуга может быть сконструирована. Например, для создания общедоступного веб-сайта электронной коммерции оператор связи может не иметь собственной IP-сети, соединенной с Интернетом. Для этого ему достаточно создать информационное наполнение сайта и поместить его на компьютере другого оператора, сеть которого имеет подключение к Интернету. Аренда физических каналов связи для создания собственной телефонной или компьютерной сети является другим типичным примером предоставления услуг при отсутствии одного из элементов аппаратно-программной инфраструктуры. Оператора, который предоставляет услуги другим операторам связи, часто называют **оператором операторов** (carrier of carriers).

В большинстве стран мира операторы связи должны получать **лицензии** от государственных органов на оказание тех или иных услуг связи. Такое положение существовало не всегда — практически во всех странах были операторы, которые являлись фактическими монополистами на рынке телекоммуникационных услуг в масштабах страны. Сегодня во многих странах мира ситуация кардинально изменилась, и процесс демонополизации телекоммуникационных услуг (прежде всего, традиционных услуг, на которых, собственно, и была установлена монополия) протекает достаточно бурно. В результате монополисты теряют свои привилегии, а иногда и принудительно разукрупняются.

Читайте «Демонополизация рынка телекоммуникационных услуг в США» на сайте www.olifer.co.uk.

Территория покрытия

По степени покрытия территории, на которой предоставляются услуги, операторы делятся на локальных, региональных, национальных и транснациональных.

Локальный оператор работает на территории города или сельского района. Традиционный локальный оператор владеет всей соответствующей транспортной инфраструктурой: физическими каналами между помещениями абонентов (квартирами, домами и офисами) и узлом связи, автоматическими телефонными станциями (АТС) и каналами связи между телефонными станциями. Сегодня к традиционным локальным операторам добавились альтернативные (CLEC), которые часто являются поставщиками услуг нового типа, прежде всего, услуг Интернета, но иногда конкурируют с традиционными операторами и в секторе телефонии.

Региональные и **национальные операторы** оказывают услуги на большой территории, располагая соответствующей транспортной инфраструктурой. Традиционные операторы этого масштаба выполняют транзитную передачу телефонного трафика между телефонными станциями локальных операторов, имея в своем распоряжении крупные транзитные АТС, связанные высокоскоростными физическими каналами связи. Это — операторы операторов, их клиентами являются, как правило, локальные операторы или крупные предприятия, имеющие отделения и филиалы в различных городах региона или страны. Располагая развитой транспортной инфраструктурой, такие операторы обычно оказывают услуги дальней связи, передавая транзитом большие объемы информации без какой-либо обработки.

Транснациональные операторы оказывают услуги в нескольких странах. Примерами таких операторов являются Cable & Wireless, Global One, Infonet. Они имеют собственные магистральные сети, покрывающие иногда несколько континентов. Часто подобные операторы тесно сотрудничают с национальными операторами, используя их сети доступа для доставки информации клиентам.

Взаимоотношения между операторами связи различного типа

Взаимосвязи между операторами различного типа (а также их сетями) иллюстрирует рис. 5.3. На рисунке показаны клиенты двух типов — индивидуальные и корпоративные. Нужно иметь в виду, что каждый клиент обычно нуждается в услугах двух видов — телефонных и передачи данных. Индивидуальные клиенты имеют в своих домах или квартирах, как правило, телефон и компьютер, а у корпоративных клиентов имеются соответствующие сети — телефонная, поддерживаемая офисным телефонным коммутатором (PBX), и локальная сеть передачи данных, построенная на собственных коммутаторах.

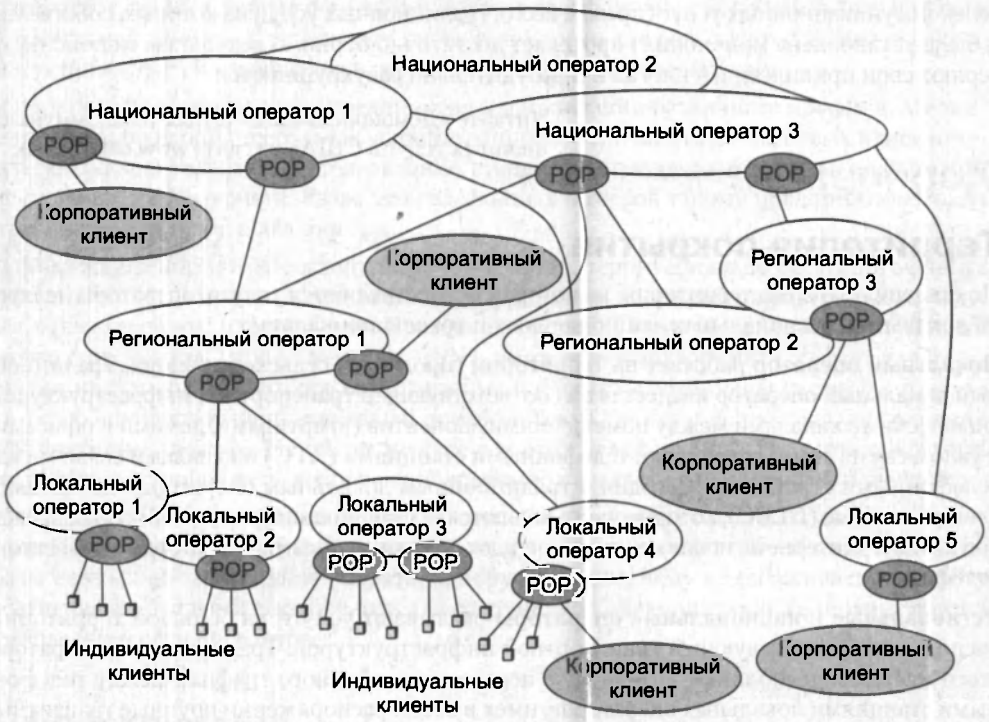


Рис. 5.3. Взаимоотношения между операторами связи различного типа

Для подключения оборудования клиентов операторы связи организуют, так называемые, **точки присутствия** (Point Of Presents, POP) — здания или помещения, в которых размещается оборудование доступа, способное подключить большое количество каналов связи, идущих от клиентов. Иногда такую точку называют **центральным офисом** (Central Office, CO) — это традиционное название для операторов телефонных сетей. К POP локальных

операторов подключаются абоненты, а к POP операторов верхних уровней — операторы нижних уровней или крупные корпоративные клиенты, которым необходимы высокие скорости доступа и большая территория покрытия, способная объединить их офисы и отделения в разных городах и странах.

Так как процесс конвергенции пока еще не привел нас к появлению единой сети для всех видов трафика, то за каждым овалом, представляющим на этом рисунке сети операторов, стоят две сети — телефонная и передачи данных.

Как видно из рисунка, в современном конкурентном телекоммуникационном мире нет строгой иерархии операторов, взаимосвязи между ними и их сетями могут быть достаточно сложными и запутанными. Например, сеть локального оператора 5 имеет непосредственную связь не только с сетью регионального оператора 3, как того требует иерархия, но и непосредственную связь с национальным оператором 3 (возможно, этот оператор предлагает более дешевые услуги по передаче международного трафика, чем это делает региональный оператор 3). Некоторые операторы могут не иметь собственной транспортной инфраструктуры (на рисунке это локальный оператор 1). Как это часто бывает в таких случаях, локальный оператор 1 предоставляет только дополнительные информационные услуги, например предлагает клиентам локального оператора 2 видео по требованию или разработку и поддержание их домашних страниц в Интернете. Свое оборудование (например, видеосервер) такой оператор часто размещает в POP другого оператора, как это и показано в данном случае.

Корпоративные сети

Корпоративная сеть — это сеть, главным назначением которой является поддержание работы конкретного предприятия, владеющего данной сетью. Пользователями корпоративной сети являются только сотрудники данного предприятия.

В отличие от сетей операторов связи, корпоративные сети, в общем случае, не оказывают услуг сторонним организациям или пользователям.

Хотя формально корпоративной сетью является сеть предприятия любого масштаба, обычно так называют сеть крупного предприятия, имеющего отделения в различных городах и, возможно, разных странах. Поэтому корпоративная сеть является составной сетью, включающей как локальные, так и глобальные сети.

Структура корпоративной сети в целом соответствует обобщенной структуре рассмотренной ранее телекоммуникационной сети. Однако имеются и отличия. Например, локальные сети, объединяющие конечных пользователей, здесь включаются в состав корпоративной сети. Кроме того, названия структурных единиц корпоративной сети отражают не только территорию покрытия, но и организационную структуру предприятия. Так, принято делить корпоративную сеть на сети отделов и рабочих групп, сети зданий и кампусов, магистраль.

Сети отделов

Сети отделов — это сети, которые используются сравнительно небольшой группой сотрудников, работающих в одном отделе предприятия. Эти сотрудники решают некоторые общие задачи, например ведут бухгалтерский учет или занимаются маркетингом. Считается, что отдел может насчитывать до 100–150 сотрудников. Сеть отдела — это локальная

сеть, которая охватывает все помещения, принадлежащие отделу. Это могут быть несколько комнат или этаж здания.

Главным назначением сети отдела является разделение локальных ресурсов, таких как приложения, данные, лазерные принтеры и модемы. Обычно сети отделов не делят на подсети и в их состав входят один или два файловых сервера и не более тридцати пользователей (рис. 5.4). В этих сетях локализуется большая часть трафика предприятия. Сети отделов обычно создаются на основе какой-либо одной сетевой технологии — Ethernet (или несколько технологий из семейства Ethernet — Ethernet, Fast Ethernet, реже Gigabit Ethernet), Token Ring или FDDI. Для такой сети характерен один или максимум два типа операционных систем.

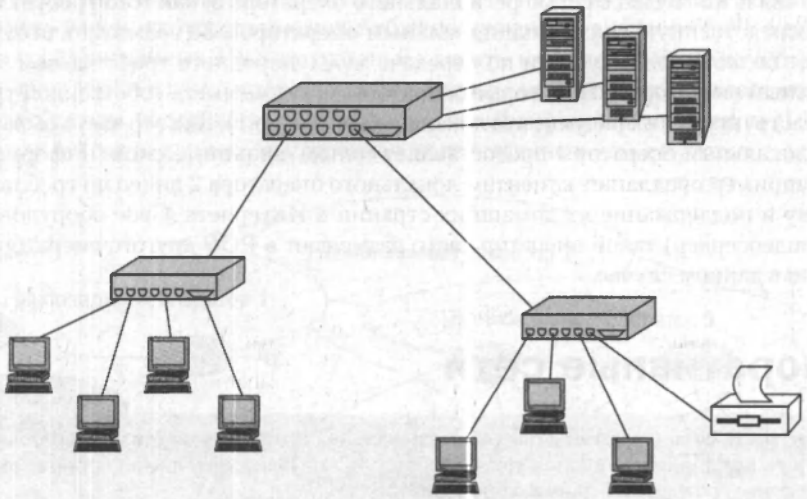


Рис. 5.4. Пример сети масштаба отдела

Задачи сетевого администрирования на уровне отдела относительно просты: добавление новых пользователей, устранение простых отказов, установка новых узлов и новых версий программного обеспечения. Такой сетью может управлять сотрудник, посвящающий обязанностям администратора только часть своего времени. Чаще всего администратор сети отдела не имеет специальной подготовки, но является тем человеком в отделе, который лучше всех разбирается в компьютерах, и само собой получается так, что он занимается администрированием сети.

Существует и другой тип сетей, близкий к сетям отделов, — **сети рабочих групп**. К таким сетям относят совсем небольшие сети, включающие до 10–20 компьютеров. Характеристики сетей рабочих групп практически не отличаются от описанных характеристик сетей отделов. Такие свойства, как простота сети и однородность, здесь проявляются в наибольшей степени, в то время как сети отделов могут приближаться в некоторых случаях к следующему по масштабу типу сетей — сетям зданий и кампусов.

В сетях рабочих групп еще часто используются технологии локальных сетей на разделяемых средах. По мере продвижения по иерархии вверх — к сетям отделов, зданий и кампусов, разделяемые среды встречаются все реже и реже, уступая место коммутируемым сетям.

Сеть отдела может входить в состав сети здания (кампуса) или же представлять собой сеть удаленного офиса предприятия. В первом случае сеть отдела подключается к сети здания

или кампуса с помощью технологии локальной сети, которой сегодня, скорее всего, будет одна из представительниц семейства Ethernet. Во втором случае сеть удаленного офиса подключается непосредственно к магистрали сети с помощью какой-либо технологии WAN, например Frame Relay.

Сети зданий и кампусов

Сеть здания объединяет сети различных отделов одного предприятия в пределах отдельного здания, а **сеть кампуса** — одной территории (кампуса), покрывающей площадь в несколько квадратных километров. Для построения сетей зданий (кампусов) используются технологии локальных сетей, возможностей которых достаточно, чтобы обеспечить требуемую зону покрытия.

Обычно сеть здания (кампуса) строится по иерархическому принципу с собственной магистралью, построенной на базе технологии Gigabit Ethernet, к которой присоединяются сети отделов, использующие технологию Fast Ethernet или Ethernet (рис. 5.5). Магистраль Gigabit Ethernet практически всегда коммутируемая, хотя эта технология и имеет вариант на разделяемой среде.

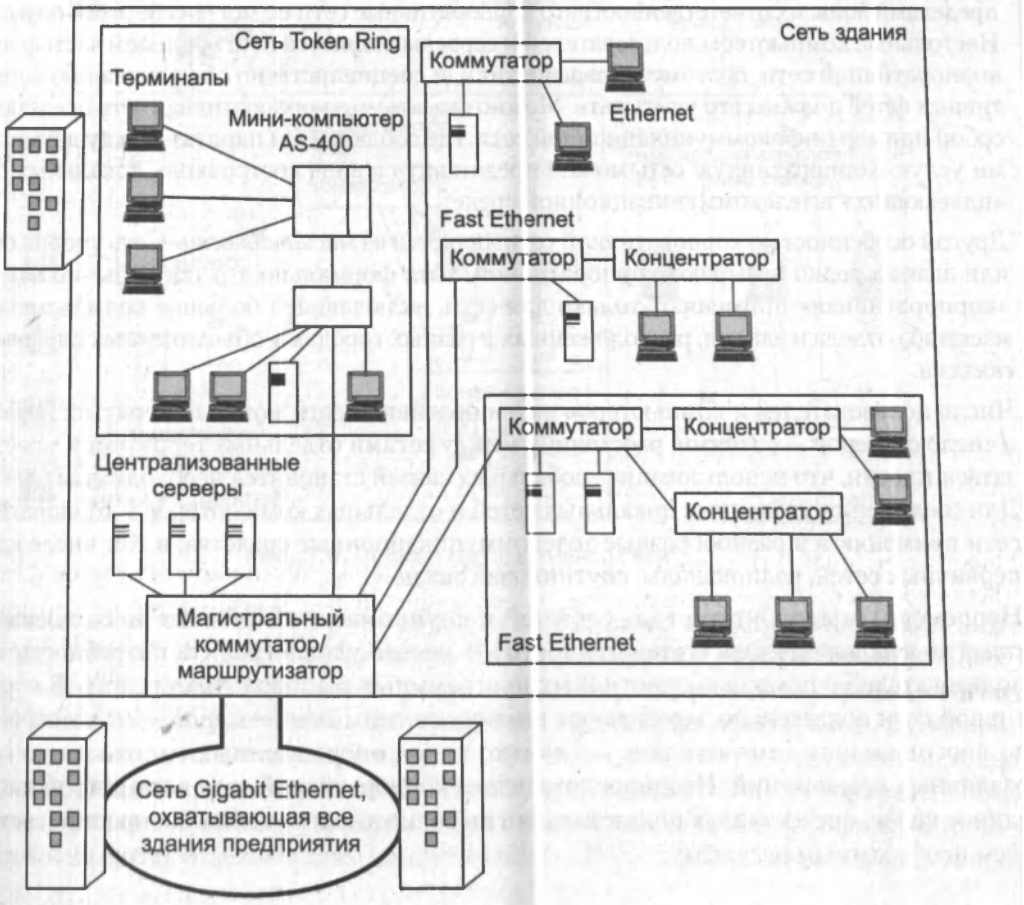


Рис. 5.5. Пример сети кампуса

Услуги такой сети включают взаимодействие между сетями отделов, доступ к общим базам данных предприятия, доступ к общим факс-серверам, высокоскоростным модемам и высокоскоростным принтерам. В результате сотрудники каждого отдела предприятия получают доступ к некоторым файлам и ресурсам сетей других отделов. Важной услугой, предоставляемой сетями кампусов, является доступ к корпоративным базам данных независимо от того, на каких типах компьютеров эти базы располагаются.

Именно на уровне сети кампуса возникают проблемы интеграции неоднородного аппаратного и программного обеспечения. Типы компьютеров, сетевых операционных систем, сетевого аппаратного обеспечения могут отличаться в каждом отделе. Отсюда вытекают сложности управления сетями кампусов. А поскольку сети отделов, входящие в сети кампуса, достаточно независимы и часто построены на базе различных технологий, объединяющей технологией обычно является IP.

Сети масштаба предприятия

Сети масштаба предприятия, или **корпоративные сети**, отличаются тем, что в них на первый план выходят *информационные услуги*. Если сети операторов связи могут и не предоставлять информационных услуг, так как компьютеры пользователей находятся за пределами зоны их ответственности, то корпоративные сети не могут себе этого позволить. Настольные компьютеры пользователей и серверы являются неотъемлемой частью любой корпоративной сети, поэтому и разработчики, и специалисты по обслуживанию корпоративных сетей должны это учитывать. Можно сказать, что корпоративная сеть представляет собой пример инфокоммуникационной сети, где соблюдается паритет между двумя типами услуг. Корпоративную сеть можно представить в виде «островков» локальных сетей, «плавающих» в телекоммуникационной среде.

Другой особенностью корпоративной сети является ее *масштабность*. Сеть уровня отдела или здания редко называют корпоративной, хотя формально это так. Обычно название «корпоративная» применяют только для сети, включающей большое количество сетей масштаба отдела и здания, расположенных в разных городах и объединенных *глобальными связями*.

Число пользователей и компьютеров в корпоративной сети может измеряться тысячами, а число серверов — сотнями; расстояния между сетями отдельных территорий могут оказаться такими, что использование глобальных связей становится необходимым (рис. 5.6). Для соединения удаленных локальных сетей и отдельных компьютеров в корпоративной сети применяются разнообразные телекоммуникационные средства, в том числе каналы первичных сетей, радиоканалы, спутниковая связь.

Непременным атрибутом столь сложной и крупномасштабной сети является *высокая степень неоднородности* (гетерогенности) — нельзя удовлетворить потребности тысяч пользователей с помощью однотипных программных и аппаратных средств. В корпоративной сети обязательно задействуют различные типы компьютеров — от мэйнфреймов до персональных компьютеров, несколько типов операционных систем и множество различных приложений. Неоднородные части корпоративной сети должны работать как единое целое, предоставляя пользователям по возможности удобный и простой доступ ко всем необходимым ресурсам.

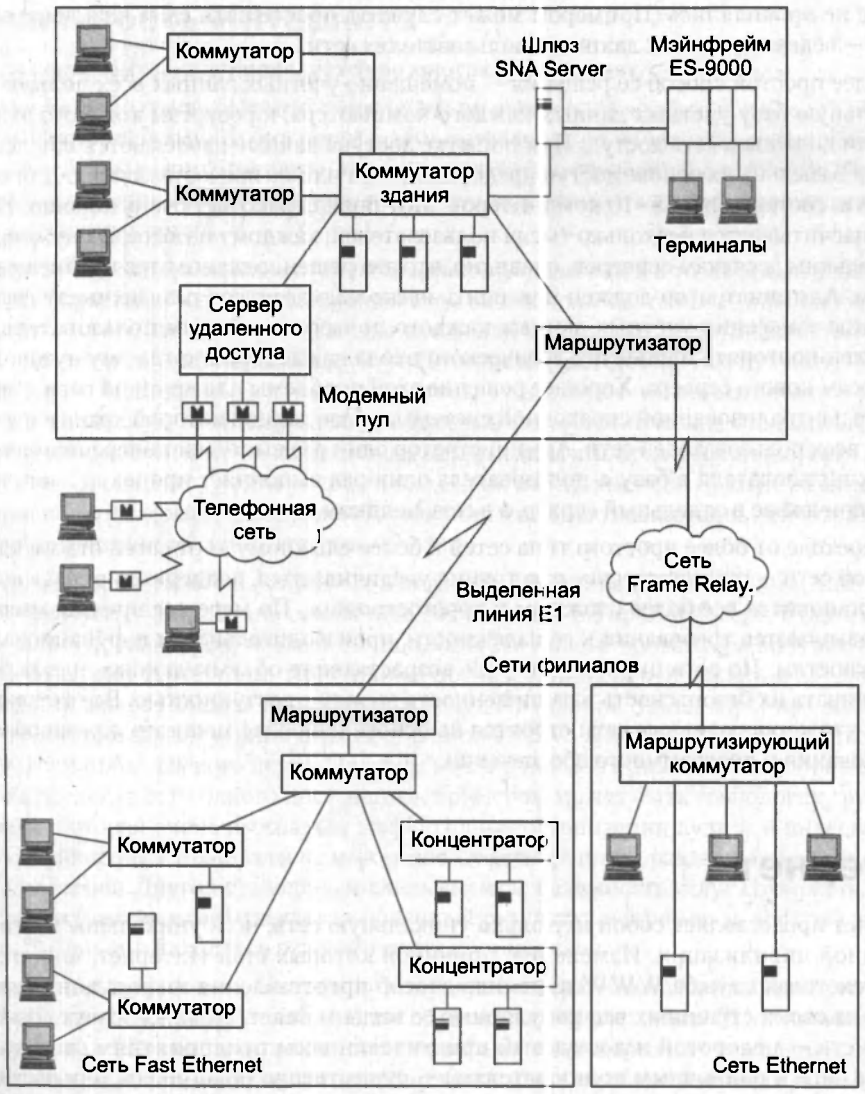


Рис. 5.6. Пример корпоративной сети

Появление корпоративных сетей — это хорошая иллюстрация известного философского постулата *о переходе количества в качество*. При объединении в единую сеть отдельных сетей крупного предприятия, имеющего филиалы в разных городах и даже странах, многие количественные характеристики объединенной сети превосходят некоторый критический порог, за которым начинается новое качество. В этих условиях существующие методы и подходы к решению традиционных задач сетей меньших масштабов для корпоративных сетей оказались непригодными. На первый план вышли такие задачи и проблемы, которые в сетях рабочих групп, отделов и даже кампусов либо имели второстепенное значение, либо

вообще не проявлялись. Примером может служить простейшая (для небольших сетей) задача — ведение учетных данных о пользователях сети.

Наиболее простой способ ее решения — помещение учетных данных всех пользователей в локальную базу учетных данных каждого компьютера, к ресурсам которого эти пользователи должны иметь доступ. При попытке доступа данные извлекаются из локальной учетной базы и на их основе доступ предоставляется или не предоставляется. Для небольшой сети, состоящей из 5–10 компьютеров, этот подход работает очень хорошо. Но если в сети насчитывается несколько тысяч пользователей, каждому из которых нужен доступ к нескольким десяткам серверов, очевидно, что это решение становится крайне неэффективным. Администратор должен повторить несколько десятков раз (по числу серверов) операцию занесения учетных данных каждого пользователя. Сам пользователь также вынужден повторять процедуру логического входа каждый раз, когда ему нужен доступ к ресурсам нового сервера. Хорошее решение этой проблемы для крупной сети — использование централизованной справочной системы, в базе данной которой хранятся учетные записи всех пользователей сети. Администратор один раз выполняет операцию занесения данных пользователя в базу, а пользователь один раз выполняет процедуру логического входа, причем не в отдельный сервер, а в сеть целиком.

При переходе от более простого типа сетей к более сложному — от сетей отдела к корпоративной сети — географические расстояния увеличиваются, поддержание связи компьютеров становится все более сложным и дорогостоящим. По мере увеличения масштабов сети повышаются требования к ее надежности, производительности и функциональным возможностям. По сети циркулируют все возрастающие объемы данных, и сеть должна обеспечивать их безопасность и защищенность наряду с доступностью. Все это приводит к тому, что корпоративные сети строятся на основе наиболее мощного и разнообразного оборудования и программного обеспечения.

Интернет

Интернет представляет собой не только уникальную сеть, но и уникальное явление современной цивилизации. Изменения, причиной которых стал Интернет, многогранны. Гипертекстовая служба WWW изменила способ представления информации человеку, собрав на своих страницах все популярные ее виды — текст, графику и звук. Транспорт Интернета — недорогой и доступный практически всем предприятиям (а через телефонные сети и одиночным пользователям) — существенно облегчил задачу построения территориальной корпоративной сети, одновременно выдвинув на первый план проблему защиты корпоративных данных при их передаче через в высшей степени общедоступную сеть с многомиллионным «населением». Стек TCP/IP, на котором строится Интернет, стал самым популярным.

Интернет неуклонно движется к тому, чтобы стать общемировой сетью интерактивного взаимодействия людей. Он начинает все больше и больше использоваться не только для распространения информации, в том числе рекламной, но и для осуществления самих деловых операций — покупки товаров и услуг, перемещения финансовых активов и т. п. Это в корне меняет для многих предприятий саму канву ведения бизнеса, поскольку изменяет поведение клиентов, значительная часть которых предпочитает совершать электронные сделки.

Уникальность Интернета

Уникальность Интернета проявляется во многих отношениях.

Прежде всего, это *самая большая в мире сеть*: по числу пользователей, по территории покрытия, по суммарному объему передаваемого трафика, по количеству входящих в ее состав сетей. Темпы роста Интернета, хотя и снизились по сравнению с периодом Интернет-революции середины 90-х годов, остаются очень высокими и намного превышают темпы роста телефонных сетей.

Интернет — это *сеть, не имеющая единого центра управления* и в то же время работающая по единым правилам и предоставляющая всем своим пользователям единый набор услуг. Интернет — это «сеть сетей», но каждая входящая в Интернет сеть управляется независимым оператором — **поставщиком услуг Интернета** (Internet Service Provider, ISP), или **провайдером**. Некоторые центральные органы существуют, но они отвечают только за единую техническую политику, за согласованный набор технических стандартов, за централизованное назначение таких жизненно важных для гигантской составной сети параметров, как имена и адреса компьютеров и входящих в Интернет сетей, но не за ежедневное поддержание сети в работоспособном состоянии. Такая высокая степень децентрализации имеет свои достоинства и недостатки.

Достоинства проявляются, например, в легкости наращивания Интернета. Так, новому поставщику услуг достаточно заключить соглашение, по крайней мере, с одним из существующих провайдеров, после чего пользователи нового провайдера получают доступ ко всем ресурсам Интернета. Негативные последствия децентрализации заключаются в сложности модернизации технологий и услуг Интернета. Такие коренные изменения требуют согласованных усилий всех поставщиков услуг, в случае «одного собственника» они проходили бы намного легче. Недаром многие новые технологии пока применяются только в пределах сети одного поставщика, примером может быть технология групповой рассылки, которая очень нужна для эффективной организации аудио- и видеовещания через Интернет, но все еще пока не может преодолеть границы, разделяющие сети различных провайдеров. Другой пример — не очень высокая надежность услуг Интернета, так как никто из поставщиков не отвечает за конечный результат, например за доступ клиента *A* к сайту *B*, если они находятся в сетях разных поставщиков.

Интернет — *недорогая сеть*. Например, популярность сравнительно новой услуги Интернета — интернет-телефонии — во многом объясняется существенно более низкими тарифами доступа в Интернет по сравнению с тарифами традиционных телефонных сетей. За низкой стоимостью стоит не временное снижение цен в надежде завоевать новый рынок, а вполне объективная причина — более низкая стоимость транспортной инфраструктуры Интернета как сети с коммутацией пакетов по сравнению с инфраструктурой телефонных сетей. Существуют, конечно, опасения, что по мере усовершенствования технологий и услуг доступ в Интернет будет обходиться все дороже и дороже. Эту опасность осознают и разработчики технологий Интернета, и поставщики услуг, проверяя каждое нововведение и с этой позиции.

Интернет не стал бы тем, чем он стал, если бы не еще одна его уникальная черта — *необъятное информационное наполнение и простота доступа к этой информации* для всех пользователей Интернета. Мы имеем в виду те сотни тысяч терабайтов информации, которые хранятся на веб-серверах Интернета и доступны пользователям Интернета в форме веб-страниц. Удобная форма представления взаимосвязей между отдельными информаци-

онными фрагментами в виде гиперссылок и стандартный графический браузер, который одинаково просто и эффективно работает во всех популярных операционных системах, совершили революцию. Интернет стал быстро заполняться самой разнообразной информацией в форме веб-страниц, превращаясь одновременно в энциклопедию, ежедневную газету, рекламное агентство и огромный магазин. Многие люди сегодня не представляют своей жизни без регулярного использования Интернета и для переписки со знакомыми, и для поиска информации (которая, как правило, нужна срочно), и для поиска работы, и для оплаты счетов.

Структура Интернета

Стремительный рост числа пользователей Интернета, привлекаемых информацией, содержащейся на его сайтах, изменил отношение корпоративных пользователей и операторов связи к этой сети. Сегодня Интернет поддерживается практически всеми традиционными операторами связи. Кроме того, к ним присоединилось большое количество новых операторов, построивших свой бизнес исключительно на услугах Интернета. Поэтому общая структура Интернета, показанная на рис. 5.7, во многом является отражением общей структуры всемирной телекоммуникационной сети, фрагмент которой мы уже рассматривали на рис. 5.3.

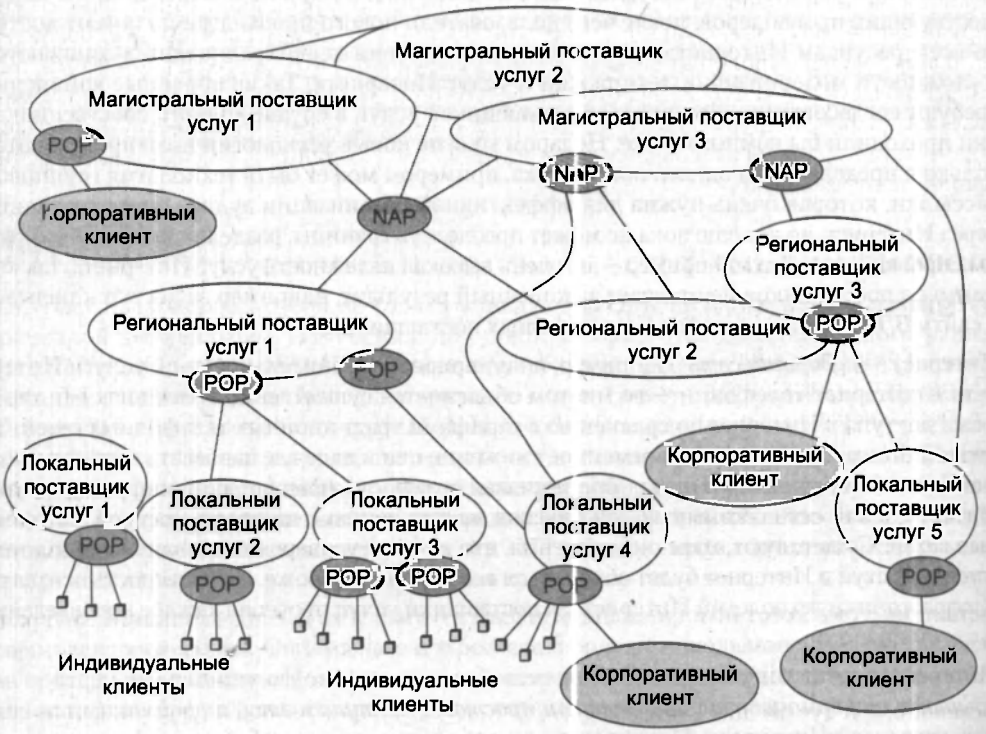


Рис. 5.7. Структура Интернета

Магистральные поставщики услуг являются аналогами транснациональных операторов связи. Они обладают собственными транспортными магистралями, покрывающими крупные регионы (страна, континент, весь земной шар). Примерами магистральных поставщиков услуг являются такие компании, как Cable & Wireless, WorldCom, Global One.

Соответственно, **региональные поставщики услуг** оказывают услуги Интернета в рамках определенного региона (штат, графство, округ — в зависимости от принятого в той или иной стране административного деления), а **локальные поставщики услуг** работают, как правило, в пределах одного города.

Связи между поставщиками услуг строятся на основе двухсторонних коммерческих соглашений о взаимной передаче трафика. Такие соглашения называют **пиринговыми** (от англ. peering — соседственный). Магистральный оператор обычно имеет пиринговые соглашения со всеми остальными магистральными операторами (так как их немного), а региональные операторы, как правило, заключают такие соглашения с одним из магистральных операторов и с несколькими другими региональными операторами.

Для того чтобы провайдерам было проще организовывать свои пиринговые связи, в Интернете существуют специальные **центры обмена трафиком**, в которых соединяются сети большого количества провайдеров. Такие центры обмена обычно называются Internet eXchange (IX), или Network Access Point (NAP).

Центр обмена трафиком является средством реализации пиринговых связей, для этого он предоставляет поставщикам услуг помещение и стойки для установки коммутационного оборудования. Все физические и логические соединения между своим оборудованием поставщики услуг выполняют самостоятельно. Это значит, что не все сети провайдеров, которые пользуются услугами некоторого центра обмена данными, автоматически обмениваются трафиком друг с другом, обмен происходит между сетями только в том случае, когда между провайдерами заключено пиринговое соглашение и они его реализовали в данном центре обмена.

Классификация провайдеров Интернета по видам оказываемых услуг

Общий термин **провайдер**, или **поставщик услуг**, **Интернета** (Internet Service Provider, ISP) обычно относят к компаниям, которые выполняют для конечных пользователей лишь *транспортную* функцию — обеспечивают передачу их трафика в сети других поставщиков.

Поставщиком интернет-контента (Internet Content provider, ICP) называют такого провайдера, который имеет собственные информационно-справочные ресурсы, предоставляя их содержание — **контент** (content) — в виде веб-сайтов. Многие поставщики услуг Интернета являются одновременно поставщиками интернет-контента.

Поставщик услуг хостинга (Hosting Service Provider, HSP) — это компания, которая предоставляет свое помещение, свои каналы связи и серверы для *размещения контента*, созданного другими предприятиями.

Поставщики услуг по доставке контента (Content Delivery Provider, CDP) — это предприятия, которые не создают информационного наполнения, а занимаются *доставкой контента* в многочисленные точки доступа, максимально приближенные к пользователям, что позволяет повысить скорость доступа пользователей к информации.

Поставщики услуг по поддержке приложений (Application Service Provider, ASP) предоставляют клиентам *доступ к крупным универсальным программным продуктам*, которые самим пользователям сложно поддерживать. Обычно это корпоративные пользователи, которых интересуют приложения класса управления предприятием, такие как SAP R3.

Так как Интернет стал уже явлением социальной жизни, растет количество поставщиков, предоставляющих сугубо специализированные услуги, например **поставщики биллинговых услуг** (Billing Service Provider, BSP) *обеспечивают оплату счетов по Интернету*, сотрудничая с муниципальными службами и поставщиками тепла и электроэнергии.

Выводы

Классификация компьютерных сетей может быть выполнена на основе различных критериев. Это могут быть технологические характеристики сетей, такие как топология, метод коммутации, метод продвижения пакетов, тип используемой среды передачи. Сети классифицируют и на основе других признаков, не являющихся технологическими, таких, например, как отношение собственности (частные, государственные, общественные), тип потребителей предоставляемых услуг (сети операторов и корпоративные сети), функциональная роль (магистраль, сеть доступа).

Компьютерные сети предоставляют услуги двух типов: информационные и транспортные. Часто под термином «сетевые услуги» понимают транспортные услуги, считая, что основной функцией сети является передача информации. Информационные услуги предоставляются конечными узлами сети — серверами, а транспортные — промежуточными узлами, которыми являются коммутаторы и маршрутизаторы сети.

Компьютерную сеть можно описать с помощью обобщенной структуры, которая справедлива для любой телекоммуникационной сети. Такая обобщенная структура состоит из сетей доступа, магистрали и информационных центров.

Специализированное предприятие, которое создает телекоммуникационную сеть для оказания общедоступных услуг, владеет этой сетью и поддерживает ее работу, называется оператором связи.

Операторы связи отличаются друг от друга набором предоставляемых услуг, территорией, в пределах которой предоставляются услуги, типом клиентов, на которых ориентируются их услуги, а также имеющейся во владении оператора инфраструктурой — линиями связи, коммутационным оборудованием, информационными серверами и т. п. Операторов связи, специализирующихся на предоставлении услуг компьютерных сетей, обычно называют поставщиками услуг.

Корпоративная сеть — это сеть, главным назначением которой является поддержание работы конкретного предприятия, владеющего сетью. Пользователями корпоративной сети являются только сотрудники данного предприятия.

Интернет является уникальной компьютерной сетью, предоставляющей разнообразные услуги во всемирном масштабе.

Вопросы и задания

1. Проведите классификацию компьютерной сети вашего учебного заведения во всех аспектах, описанных в данной главе.
2. Приведите примеры информационных центров различных типов телекоммуникационных сетей.
3. Перечислите основные требования, которым должны удовлетворять сети доступа и магистральные сети.

4. Перечислите типы клиентов операторов связи.
5. Можно ли назвать сеть оператора связи корпоративной сетью?
6. Назовите основные характеристики сетей операторов связи.
7. Любые ли коммуникационные устройства, работающие в корпоративной сети, относят к классу корпоративных?
8. В чем заключается услуга по предоставлению доступа в Интернет?
9. Заполните представленную здесь таблицу, установив соответствие между описаниями сетей и их типами (один тип сети не описан).

Описание сети	Корпоративная сеть	Сеть кампуса	Сеть отдела	Сеть оператора
Сеть используется группой сотрудников до 100–150 человек				
Все сотрудники сети связаны с решением частной бизнес-задачи				
Сеть создана на основе какой-либо одной сетевой технологии				
Сеть включает тысячи пользовательских компьютеров, сотни серверов				
Сеть обладает высокой степенью гетерогенности компьютеров, коммуникационного оборудования, операционных систем и приложений				
В сети используются глобальные связи				
Сеть объединяет более мелкие сети в пределах отдельного здания или одной территории				
Глобальные соединения в сети не используются				
Службы сети предоставляют всем сотрудникам доступ к общим базам данных предприятия				

10. В сетях какого типа, корпоративных или ISP-сетях, доля локальных сетей больше?
11. В чем состоит уникальность Интернета?
12. Назовите варианты специализации поставщиков услуг Интернета.
13. Опишите последовательность необходимых с вашей точки зрения действий руководства предприятия для того, чтобы это предприятие могло стать поставщиком услуг Интернета и начать предоставлять услуги клиентам.

ГЛАВА 6 Сетевые характеристики

Компьютерная сеть представляет собой сложную и дорогую систему, решающую ответственные задачи и обслуживающую большое количество пользователей. Поэтому очень важно, чтобы сеть не просто работала, но работала качественно.

Понятие *качества обслуживания* можно трактовать очень широко, включая в него все возможные и желательные для пользователя свойства сети и поставщика услуг, поддерживающего работу этой сети. Для того чтобы пользователь и поставщик услуг могли более конкретно обсуждать проблемы обслуживания и строить свои отношения на формальной основе, существует ряд общепринятых характеристик качества предоставляемых сетью услуг. Мы будем рассматривать в этой главе только характеристики качества *транспортных* услуг сети, которые намного проще поддаются формализации, чем характеристики качества информационных услуг. Характеристики качества транспортных услуг отражают такие важнейшие свойства сети, как производительность, надежность и безопасность.

Часть этих характеристик может быть оценена количественно и измерена при обслуживании пользователя. Пользователь и поставщик услуг могут заключить соглашение об уровне обслуживания, в котором оговорить требования к количественным значениям некоторых характеристик, например, к доступности предоставляемых услуг.

Термин «качество обслуживания» часто употребляется в узком смысле, как одно из современных направлений в сетевых технологиях, цель которого состоит в разработке методов качественной передачи трафика через сеть. Характеристики качества обслуживания объединяет то, что все они отражают отрицательное влияние механизма очередей на передачу трафика.

Это влияние, в частности, может выражаться во временном снижении скорости передачи трафика, доставке пакетов с переменными задержками и потере пакетов из-за перегрузки буферов коммутаторов.

Типы характеристик

Субъективные оценки качества

Если опросить пользователей, чтобы выяснить, что они вкладывают в понятие качественных сетевых услуг, то можно получить очень широкий спектр ответов. Среди них, скорее всего, встретятся следующие мнения:

- сеть работает быстро, без задержек;
- трафик передается надежно;
- услуги предоставляются бесперебойно по схеме 24 × 7 (то есть 24 часа в сутки семь дней в неделю);
- служба поддержки работает хорошо, давая полезные советы и помогая разрешить проблемы;
- услуги предоставляются по гибкой схеме, мне нравится, что можно в любой момент и в широких пределах повысить скорость доступа к сети и увеличить число точек доступа;
- поставщик не только передает мой трафик, но и защищает мою сеть от вирусов и атак злоумышленников;
- я всегда могу проконтролировать, насколько быстро и без потерь сеть передает мой трафик;
- поставщик предоставляет широкий спектр услуг, в частности помимо стандартного доступа в Интернет он предлагает хостинг для моего персонального веб-сайта и услуги IP-телефонии.

Эти субъективные оценки отражают *пожелания пользователей* к качеству сетевых сервисов. Пользователи, клиенты — это важнейшая сторона любого бизнеса, в том числе бизнеса сетей передачи данных, но существует и еще одна сторона — *поставщик услуг* (коммерческий, если это публичная сеть, и некоммерческий, если это корпоративная сеть). Для того чтобы пользователи и поставщики услуг могли обоснованно судить о качестве сервисов, существуют *формализованные характеристики качества сетевых услуг*, которые позволяют количественно оценить тот или иной аспект качества.

Характеристики и требования к сети

Работая в сети, пользователь формулирует определенные *требования* к ее характеристикам. Например, пользователь может потребовать, чтобы средняя скорость передачи его информации через сеть не опускалась ниже 2 Мбит/с. То есть в данном случае пользователь задает тот диапазон значений для средней скорости передачи информации через сеть, который для него означает хорошее качество сервиса.

Все множество характеристик качества транспортных услуг сети можно отнести к одной из следующих групп:

- производительность;
- надежность;
- безопасность;
- характеристики, имеющие значение только для поставщика услуг.

Первые три группы соответствуют трем наиболее важным для пользователя характеристикам транспортных услуг — возможности без потерь и перерывов в обслуживании (**надежность**) передавать с заданной скоростью (**производительность**) защищенную от несанкционированного доступа и подмены информацию (**безопасность**[1]). Понятно, что поставщик сетевых услуг, стремясь удовлетворить требования пользователей, также уделяет внимание этим характеристикам. В то же время существует ряд важных для поставщика характеристик сети, которые не интересуют пользователей.

Дело в том, что сеть обслуживает большое количество пользователей, и поставщику услуг нужно организовать работу своей сети таким образом, чтобы одновременно удовлетворить требования *всех* пользователей. Как правило, это сложная проблема, так как основные ресурсы сети — линии связи и коммутаторы (маршрутизаторы) — разделяются между информационными потоками пользователей. Поставщику необходимо найти такой баланс в распределении ресурсов между конкурирующими потоками, чтобы требования всех пользователей были соблюдены. Решение этой задачи включает *планирование* и *контроль* расходования ресурсов в процессе передачи пользовательского трафика. Поставщика интересуют те характеристики ресурсов, с помощью которых он обслуживает пользователей. Например, его интересует производительность коммутатора, так как поставщик должен оценить, какое количество потоков пользователей он может обработать с помощью данного коммутатора. Для пользователя производительность коммутатора не представляет интерес, ему важен конечный результат — будет его поток обслужен качественно или нет.

Временная шкала

Рассмотрим еще один способ классификации характеристик — в соответствии с временной шкалой, на которой эти характеристики определяются.

Долговременные характеристики определяются на промежутках времени от нескольких месяцев до нескольких лет. Их можно назвать характеристиками проектных решений. Примерами таких характеристик являются набор моделей и количество коммутаторов в сети, топология и пропускная способность линий связи. Эти параметры сети прямо влияют на характеристики качества услуг сети. Одно проектное решение может оказаться удачным и сбалансированным, так что потоки трафика не будут испытывать перегрузок; другое может создавать узкие места для потоков, в результате задержки и потери пакетов превысят допустимые пределы. Понятно, что полная замена или глубокая модернизация сети связана с большими затратами финансовых средств и времени, поэтому они происходят не слишком часто и продолжают оказывать влияние на качество сети в течение продолжительного времени.

Среднесрочные характеристики определяются на интервалах времени от нескольких секунд до нескольких дней, как правило, включающих обслуживание большого количества пакетов. Например, к среднесрочным характеристикам может быть отнесено усредненное значение задержки пакетов по выборке, взятой в течение суток.

Краткосрочные характеристики относятся к темпу обработки отдельных пакетов и измеряются в микросекундном и миллисекундном диапазонах. Например, время буферизации, или время пребывания пакета в очереди коммутатора или маршрутизатора, является характеристикой этой группы. Для анализа и обеспечения требуемого уровня краткосрочных характеристик разработано большое количество методов, получивших название методов **контроля и предотвращения перегрузок** (congestions control and congestion avoidance).

---

[1] Вопросы безопасности компьютерных сетей обсуждаются в главе 24

Соглашение об уровне обслуживания

Естественной основой нормального сотрудничества поставщика услуг и пользователей является *договор*. Договор всегда заключается между клиентами и поставщиками услуг публичных сетей передачи данных, однако не всегда в нем указываются количественные требования к эффективности предоставляемых услуг. Очень часто в договоре услуга специфицируется очень общо, например «предоставление доступа в Интернет».

Однако существует и другой тип договора, называемый **соглашением об уровне обслуживания** (Service Level Agreement, SLA). В таком соглашении поставщик услуг и клиент описывают качество предоставляемой услуги в количественных терминах, пользуясь характеристиками эффективности сети. Например, в SLA может быть записано, что поставщик обязан передавать трафик клиента без потерь и с той средней скоростью, с которой пользователь направляет его в сеть. При этом оговорено, что это соглашение действует только в том случае, если средняя скорость трафика пользователя не превышает, например, 3 Мбит/с, в противном случае поставщик получает право просто не передавать избыточный трафик. Для того чтобы каждая сторона могла контролировать соблюдение этого соглашения, необходимо еще указать период времени, на котором будет измеряться средняя скорость, например день, час или секунда. Еще более определенным соглашение SLA становится в том случае, когда в нем указываются средства и методы измерения характеристик сети, чтобы у поставщика и пользователя не было расхождений при контроле соглашения.

Соглашения SLA могут заключаться не только между поставщиками коммерческих услуг и их клиентами, но и между подразделениями одного и того же предприятия. В этом случае поставщиком сетевых услуг может являться, например, отдел информационных технологий, а потребителем — производственный отдел.

Производительность

Мы уже знакомы с такими важными долговременными характеристиками производительности сетевых устройств, как пропускная способность каналов или производительность коммутаторов и маршрутизаторов. Наибольший интерес данные характеристики представляют для поставщиков услуг — на их основе поставщик услуг может планировать свой бизнес, рассчитывая максимальное количество клиентов, которое он может обслужить, определяя рациональные маршруты прохождения трафика и т. п.

Однако клиента интересуют другие характеристики производительности, которые позволят ему количественно оценить, насколько быстро и качественно сеть передает его трафик. Для того чтобы определить эти характеристики, воспользуемся моделью идеальной сети.

Идеальная сеть

В разделе «Количественное сравнение задержек» главы 3 мы рассмотрели различные составляющие задержек в сети с коммутацией пакетов. Напомним, что такими составляющими являются показатели времени:

- передачи данных в канал (время сериализации);
- распространения сигнала;

- ожидания пакета в очереди;
- коммутации пакета.

Два первых типа задержки определяются свойствами каналов передачи данных (битовой скоростью и скоростью распространения сигнала в среде) и являются фиксированными для пакета фиксированной длины.

Две вторых составляющих зависят от характеристик сети коммутации пакетов и в общем случае являются переменными.

Будем считать, что сеть с коммутацией пакетов работает идеально, если она передает каждый бит информации с постоянной скоростью, равной скорости распространения света в физической среде. Другими словами, идеальная сеть с коммутацией пакетов не вносит никаких дополнительных задержек в передачу данных помимо тех, которые вносятся каналами связи (и работает в отношении временных характеристик передачи данных так, как если бы она была сетью с коммутацией каналов).

Результат передачи пакетов такой идеальной сетью иллюстрирует рис. 6.1. На верхней оси показаны значения времени поступления пакетов в сеть от узла отправителя, а на нижнем — значения времени поступления пакетов в узел назначения. Другими словами, можно сказать, что верхняя ось показывает **предложенную нагрузку** сети, а нижняя — результат передачи этой нагрузки через сеть.

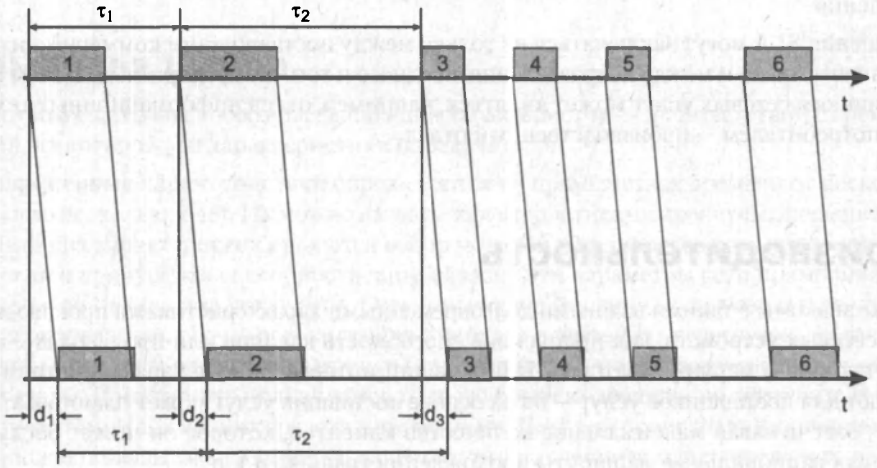


Рис. 6.1. Передача пакетов идеальной сетью

Пусть задержка передачи пакета определяется как интервал времени между моментом отправления первого бита пакета в канал связи узлом отправления и моментом поступления первого бита пакета в узел назначения соответственно (на рисунке обозначены задержки d_1, d_2 и d_3 пакетов 1, 2 и 3 соответственно).

Как видно из рисунка, идеальная сеть доставляет все пакеты узлу назначения:

- не потеряв ни один из них (и не исказив информацию ни в одном из них);
- в том порядке, в котором они были отправлены;
- с одной и той же и минимально возможной задержкой ($d_1 = d_2$ и т. д.).

Важно, что все интервалы между соседними пакетами сеть сохраняет в неизменном виде. Например, если интервал между первым и вторым пакетами составляет при отправлении τ_1 секунд, а между вторым и третьим — τ_2, то такими же интервалы останутся в узле назначения.

Надежная доставка всех пакетов с минимально возможной задержкой и сохранением временны́х интервалов между ними удовлетворит любого пользователя сети независимо от того, трафик какого приложения он передает по сети — веб-сервиса или IP-телефонии.

Существуют и другие определения времени задержки пакета. Например, эту величину можно определить как время между моментом отправления первого бита пакета в канал связи узлом отправления и моментом поступления последнего бита пакета в узел назначения соответственно. Нетрудно видеть, что в этом определении в задержку пакета включено время сериализации, кроме того, понятно, что оба определения не противоречат друг другу и величина задержки, полученная в соответствии с одним определением, легко преобразуется в величину задержки, полученной в соответствии с другим. Мы выбрали первое определение для иллюстрации идеального поведения сети с коммутацией пакетов потому, что в этом случае задержка не зависит от размера пакета, что удобнее использовать, описывая «идеальность» обслуживания пакетов.

Теперь посмотрим, какие отклонения от идеала могут встречаться в *реальной* сети и какими характеристиками можно эти отклонения описывать (рис. 6.2).

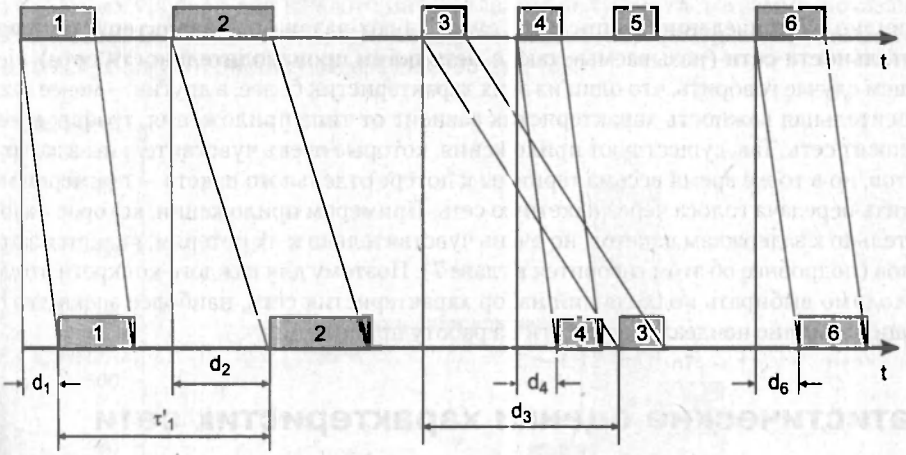


Рис. 6.2. Передача пакетов реальной сетью

Пакеты доставляются сетью узлу назначения *с различными задержками*. Как мы уже знаем, это неотъемлемое свойство сетей с коммутацией пакетов.

Случайный характер процесса образования очередей приводит к случайным задержкам, при этом задержки отдельных пакетов могут быть значительными, в десятки раз превосходя среднюю величину задержек ($d_1 \neq d_2 \neq d_3$ и т. д.). Неравномерность задержек приводит к неравномерным интервалам между соседними пакетами. То есть изменяется характер временны́х соотношений между соседними пакетами, а это может катастрофически сказаться на качестве работы некоторых приложений. Например, при цифровой передаче речи

(или более обобщенно — звука) неравномерность интервалов между пакетами, несущими замеры голоса, приводит к существенным искажениям речи.

Пакеты могут доставляться узлу назначения *не в том порядке*, в котором они были отправлены, например, на рис. 6.2 пакет 4 поступил в узел назначения раньше, чем пакет 3. Такие ситуации встречаются в дейтаграммных сетях, когда различные пакеты одного потока передаются через сеть различными маршрутами, а следовательно, ожидают обслуживания в разных очередях с разным уровнем задержек. Очевидно, что пакет 3 проходил через перегруженный узел или узлы, так что его суммарная задержка оказалась настолько большой, что пакет 4 прибыл раньше него.

Пакеты *могут теряться* в сети или же приходить в узел назначения *с искаженными данными*, что равносильно потере пакета, так как большинство протоколов не способно восстанавливать искаженные данные, а только определяет этот факт по значению контрольной последовательности кадра (Frame Check Sequence, FCS).

Пакеты также могут *дублироваться* по разным причинам, например из-за ошибочных повторных передач протоколов, обеспечивающих надежный обмен данными.

В реальной сети средняя скорость информационного потока на входе узла назначения может отличаться от средней скорости потока, направленного в сеть узлом-отправителем. Виной этому являются не задержки пакетов, а их потери[1]. Так, в примере, показанном на рис. 6.2, *средняя скорость исходящего потока снижается* из-за потери пакета 5. Чем больше потерь и искажений пакетов происходит в сети, тем ниже скорость информационного потока.

Как видно из приведенного описания, существуют различные **характеристики производительности сети** (называемые также **метриками производительности сети**). Нельзя в общем случае говорить, что одни из этих характеристик более, а другие — менее важные. Относительная важность характеристик зависит от типа приложения, трафик которого переносит сеть. Так, существуют приложения, которые очень чувствительны к задержкам пакетов, но в то же время весьма терпимы к потере отдельного пакета — примером может служить передача голоса через пакетную сеть. Примером приложения, которое мало чувствительно к задержкам пакетов, но очень чувствительно к их потерям, является загрузка файлов (подробнее об этом говорится в главе 7). Поэтому для каждого конкретного случая необходимо выбирать подходящий набор характеристик сети, наиболее адекватно отражающий влияние неидеальности сети на работу приложения.

Статистические оценки характеристик сети

Очевидно, что множество отдельных значений времени передачи каждого пакета в узел назначения дают исчерпывающую характеристику качества передачи трафика сетью в течение определенного промежутка времени. Однако это слишком громоздкая и, более того, избыточная характеристика производительности сети. Для того чтобы представить характеристики качества передачи последовательности пакетов через сеть в компактной форме, применяются *статистические методы*.

Статистические методы служат для оценки характеристик *случайных процессов*, а именно такой характер имеют процессы передачи пакетов сетью. Сами характеристики производительности сети, такие как, например, задержка пакета, являются *случайными величинами*.

---

[1] Это утверждение справедливо, когда интервал усреднения скорости существенно превышает величину максимальной задержки.

Статистические характеристики выявляют закономерности в поведении сети, которые устойчиво проявляются только на длительных периодах времени. Когда мы говорим о длительном периоде времени, то мы понимаем под этим интервал, в миллионы раз больший, чем время передачи одного пакета, которое в современной сети измеряется микросекундами. Так, время передачи пакета Fast Ethernet составляет около 100 мкс, Gigabit Ethernet — около 10 мкс, ячейки ATM — от долей микросекунды до 3 мкс (в зависимости от скорости передачи). Поэтому для получения устойчивых результатов нужно наблюдать поведение сети, по крайней мере, в течение минут, а лучше — нескольких часов.

Основным инструментом статистики является так называемая **гистограмма** распределения оцениваемой случайной величины. Рассмотрим этот инструмент на примере такой характеристики сети, как задержка пакета.

Будем считать, что нам удалось измерить задержку доставки каждого из 2600 пакетов, переданных между двумя узлами сети, и сохранить полученные результаты. Эти результаты называются **выборкой** случайной величины.

Для того чтобы получить гистограмму распределения, мы должны разбить весь диапазон измеренных значений задержек на несколько интервалов и подсчитать, сколько пакетов из нашей выборки попало в каждый интервал. Пусть все значения задержек укладываются в диапазон 20–90 мс. Разобьем его на семь интервалов по 10 мс. В каждый из этих интервалов, начиная с интервала 20–30 мс и т. д., попало 100 ($n1$), 200 ($n2$), 300 ($n3$), 300 ($n4$), 400 ($n5$), 800 ($n6$) и 500 ($n7$) пакетов соответственно. Отобразив эти числа в виде горизонтальных уровней для каждого интервала, мы получим гистограмму, показанную на рис. 6.3, которая, основываясь всего на семи числах $n1$, $n2$, ..., $n7$, дает нам компактную статистическую характеристику задержек 2600 пакетов.

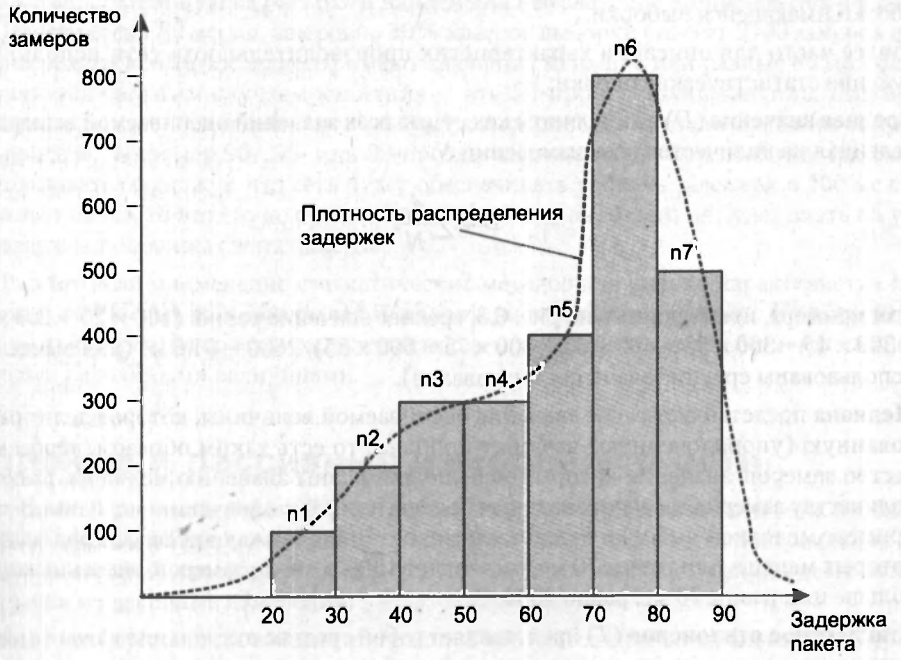


Рис. 6.3. Гистограмма распределения задержек

Гистограмма задержек дает хорошее представление о производительности сети. По ней можно судить, какие уровни задержек более вероятны, а какие — менее. Чем больше период времени, в течение которого собираются данные для построения гистограммы, тем с более высокой степенью достоверности можно предсказать поведение сети в будущем. Например, пользуясь гистограммой на рис. 6.3, можно сказать, что и в будущем при измерениях задержек пакетов у 65 % пакетов задержка не превысит 60 мс. Для получения такой оценки мы сложили общее количество пакетов, задержки которых попали во все интервалы, большие 60 мс (1700 замеров), и разделили эту величину на общее количество пакетов (2600 замеров). Другими словами, мы нашли долю пакетов, задержки которых в выборке превышают 60 мс, и считаем, что наша выборка позволяет судить о поведении сети в будущем.

Насколько точен такой прогноз? Собрали ли мы достаточно экспериментальных данных, чтобы делать более-менее достоверные прогнозы? Статистика позволяет судить и об этом, однако мы не будем рассматривать здесь эту увлекательную проблему и оставим ее специальным книгам по статистике.

При увеличении количества интервалов и времени наблюдения мы в пределе получаем непрерывную функцию, которая называется **плотностью распределения** задержки доставки пакета (показана пунктиром). В соответствии с теорией, вероятность того, что значение случайной величины окажется в определенном диапазоне, равна интегралу плотности распределения случайной величины от нижней до верхней границ данного диапазона. Таким образом, может быть вычислено вероятностное значение задержки пакета.

Гистограмма дает хорошее детальное описание соответствующей характеристики, но чаще всего используются еще более компактные **статистические оценки** характеристик, которые позволяют представить характеристику *одним числом* на основе некоторой математической обработки имеющейся выборки.

Наиболее часто для описания характеристик производительности сети используются следующие статистические оценки.

- **Среднее значение** (D) вычисляется как сумма всех значений оцениваемой величины d_i, деленная на количество всех измерений N:

$$D = \sum \frac{d_i}{N}.$$

  Для примера, приведенного на рис. 6.3, среднее значение равно: $(100 \times 25 + 200 \times 35 + 300 \times 45 + 300 \times 55 + 400 \times 65 + 800 \times 75 + 500 \times 85)/2600 = 64{,}6$ мс (для вычисления использованы средние значения интервалов).

- **Медиана** представляет такое значение оцениваемой величины, которое делит ранжированную (упорядоченную) выборку пополам, то есть таким образом, чтобы количество замеров, значения которых меньше или равны значению медианы, равнялось количеству замеров, значения которых больше или равны значению медианы. В нашем примере медианой выборки является значение 70 мс, так как число замеров, значения которых меньше или равны 70 мс, составляет 1300, а число замеров, значения которых больше или равны 70 мс, равно 1300.

- **Стандартное отклонение** (J) представляет собой среднее отклонение каждого отдельного замера от среднего значения оцениваемой величины:

$$J = \sqrt{\frac{\sum (d_i - D)^2}{N-1}}.$$

Очевидно, что если все задержки d_i равны между собой, то вариация отсутствует, что подтверждают приведенные формулы — в этом случае $D = d_i$ и $J = 0$.

- **Коэффициент вариации** — это безразмерная величина, которая равна отношению стандартного отклонения к среднему значению оцениваемой величины:

$$CV = \frac{J}{D}.$$

Коэффициент вариации характеризует оцениваемую величину без привязки к ее абсолютным значениям. Так, идеальный равномерный поток пакетов всегда будет обладать нулевым значением коэффициента вариации задержки пакета. Коэффициент вариации задержки пакета, равный 1, означает достаточно пульсирующий трафик, так как средние отклонения интервалов от некоторого среднего периода следования пакетов равны этому периоду.

- **Квантиль (процентиль)** — это такое значение оцениваемой величины, которое делит ранжированную выборку на две части так, что процент замеров, значения которых меньше или равно значению квантиля, равен некоторому заданному уровню. В этом определении фигурируют два числа: заранее заданный процент и найденное по нему и замерам выборки значение квантиля. Рассмотрим для примера выборку задержек пакетов, показанную на рис. 6.3, и найдем для нее значение 80-процентного квантиля. Ответом будет 80 мс, так как ровно 80 % замеров выборки (то есть 2100 замеров из всех интервалов кроме последнего) имеют значения, меньшие или равные 80 мс. Медиана является частным случаем квантиля — это 50-процентный квантиль. Для оценки характеристик сети обычно используют квантили с достаточно большим значением процента, например 90-, 95- или 99-процентные квантили. Это понятно, так как если пользователю скажут, что сеть будет обеспечивать уровень задержек в 100 мс с вероятностью 0,5, то это его не очень обрадует, так как он ничего не будет знать об уровне задержек половины своих пакетов.

Мы рассмотрели применение статистических методов для оценки характеристик производительности сети на примере такой характеристики, как задержка. Естественно, эти методы применяются ко всем характеристикам производительности сети, так как все они являются случайными величинами.

Активные и пассивные измерения в сети

Для того чтобы оценить некоторую характеристику производительности сети, необходимо провести определенные измерения на последовательности пакетов, поступающих на некоторый интерфейс сетевого устройства. Существует два типа измерений в сети: активные измерения и пассивные измерения.

Активные измерения основаны на генерации в узле-источнике специальных «измерительных» пакетов. Эти пакеты должны пройти через сеть тот же путь, что и пакеты, характе-

ристики которых мы собираемся оценивать. Измерения в узле назначения проводятся на последовательности «измерительных» пакетов.

Рисунок 6.4 иллюстрирует идею активных измерений. Пусть мы хотим измерить задержки пакетов некоторого приложения *A*, которые передаются от компьютера-клиента приложения *A* компьютеру-серверу приложения *A* через сеть. Вместо того чтобы пытаться измерить задержки пакетов, генерируемых клиентским компьютером, мы устанавливаем в сети два дополнительных компьютера: сервер-генератор и сервер-измеритель. Сервер-генератор генерирует измерительные пакеты (показанные на рисунке серым цветом), а сервер-измеритель измеряет задержки этих пакетов. Для того чтобы измеряемые значения были близки к значениям задержки пакетов приложения *A*, нужно, чтобы измерительные пакеты проходили через сеть по тому же пути, что и пакеты приложения *A*, то есть нужно постараться подключить сервер-генератор и сервер-измеритель по возможности ближе к оригинальным узлам. В нашем примере такое приближение достигнуто за счет подключения дополнительных узлов к портам тех же коммутаторов *S1* и *S2*, к которым подключены оригинальные узлы. Кроме того, нужно, чтобы измерительные пакеты как можно больше «походили» на оригинальные пакеты — размерами, признаками, помещенными в заголовки пакетов. Это требуется для того, чтобы сеть обслуживала их так же, как оригинальные пакеты.

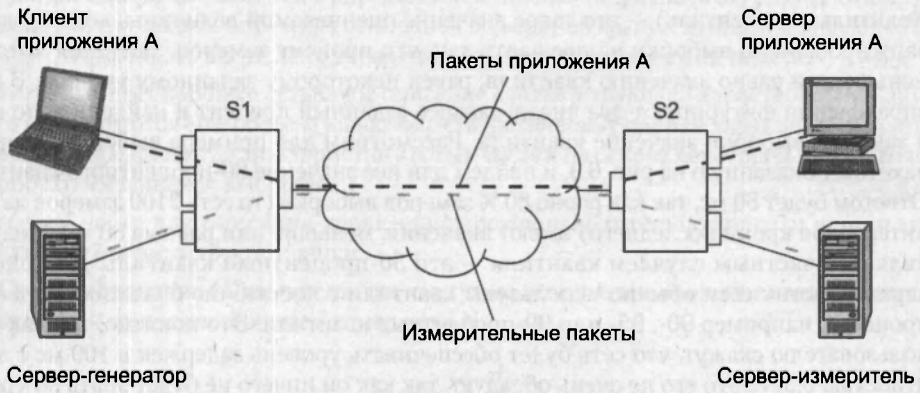


Рис. 6.4. Схема активных измерений

Однако измерительные пакеты не должны генерироваться слишком часто, иначе нагрузка сети может существенно измениться, и результаты замеров будут отличаться от тех, которые были бы получены в отсутствии измерительных пакетов. Другими словами, измерения не должны менять условий работы сети. Обычно интенсивность генерации измерительных пакетов не превосходит 20–50 пакетов в секунду. Существует специальное программное обеспечение, которое генерирует измерительные пакеты и измеряет их характеристики по прибытию на сервер-измеритель.

Возникает естественный вопрос: зачем нужно решать столько проблем: размещать дополнительное оборудование, создавать условия для измерительных пакетов, близкие к условиям обработки оригинальных пакетов, и в то же время стараться не изменить нагрузку сети? Не проще ли измерять параметры реальных пакетов? Ответ заключается в том, что активная схема упрощает процесс проведения измерений и позволяет добиться их высокой

точности. Так как сервер-генератор создает измерительные пакеты, то он легко может использовать специальный формат пакетов для того, чтобы поместить в них необходимую для измерения информацию, например временную отметку (time-stamp) отправки пакета. Затем сервер-измеритель использует эту временную отметку для вычисления времени задержки. Очевидно, что для того чтобы измерения задержки были точными, нужна хорошая синхронизация сервера-генератора и сервера-измерителя. Так как в схеме активных измерений они представляют собой выделенные узлы, такой синхронизации добиться проще, чем в случае синхронизации клиента и сервера приложения *A*, которые чаще всего представляют собой обычные компьютеры. Кроме того, иногда у инженеров, проводящих измерения, просто нет доступа к компьютерам, на которых работают приложения, чтобы установить там программное обеспечение для требуемых измерений поступающих пакетов. А если такой доступ и существует, то операционные системы клиента и сервера и их аппаратная платформа, скорее всего, не оптимизированы для точных измерений временных интервалов, а значит, вносят большие искажения в результаты (например, за счет задержек программы измерений в очереди к центральному процессору).

Однако преимущества активной схемы измерений не являются абсолютными. В некоторых ситуациях более предпочтительной является схема пассивных измерений.

Пассивные измерения основаны на измерениях характеристик реального трафика. Эту схему иллюстрирует рис. 6.5.

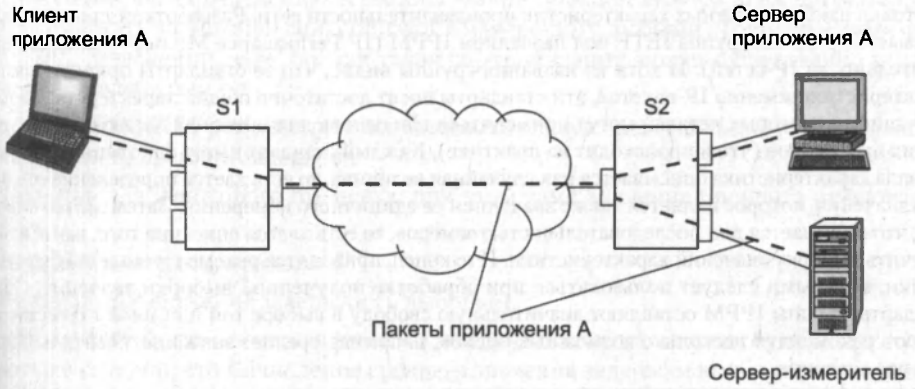


Рис. 6.5. Схема пассивных измерений

Приводя аргументы в пользу схемы активных измерений, мы, в сущности, описали проблемы, которые приходится решать при использовании схемы пассивных измерений: сложности синхронизации клиента и сервера, дополнительные и неопределенные задержки, вносимые универсальными мультпрограммными операционными системами этих компьютеров, отсутствие в заголовке используемых приложением пакетов поля для переноса по сети временной отметки.

Частично эти проблемы решаются за счет использования отдельного сервера-измерителя. Этот сервер принимает тот же входной поток пакетов, что и один из узлов, участвующий в обмене пакетами, характеристики которых нужно измерить (на рисунке показан случай, когда сервер-измеритель ставится в параллель с сервером приложения *A*). Для того чтобы сервер-измеритель получал тот же входной поток пакетов, что и оригинальный узел,

обычно прибегают к дублированию измеряемого трафика на порт, к которому подключен сервер-измеритель. Такую функцию, называемую зеркализацией портов, поддерживают многие коммутаторы локальных сетей. Сервер-измеритель может работать под управлением специализированной операционной системы, оптимизированной для выполнения точных измерений временны́х интервалов.

Сложнее решить проблему синхронизации. Некоторые протоколы переносят временны́е отметки в своих служебных полях, так что если, например, приложение *A* использует такой протокол, то часть проблемы решается. Однако и в этом случае остается открытым вопрос о точности системного времени в компьютере клиента приложения *A*; скорее всего она невысока. Поэтому в пассивном режиме измеряют те характеристики, которые не требуют синхронизации передатчика и приемника, например оценивают долю потерянных пакетов.

Возможным вариантом пассивной схемы измерений является отсутствие выделенного сервера-измерителя. Некоторые приложения сами выполняют измерения задержек поступающих пакетов, например такими функциями обладают многие приложения IP-телефонии и видеоконференций, так как информация о задержках пакетов помогает определить возможную причину неудовлетворительного качества работы приложения.

СТАНДАРТЫ ИЗМЕРЕНИЙ

Как и в любой области, в сфере измерений имеются стандарты, создающие основу для одинаковой трактовки наиболее важных характеристик производительности сети. Разработкой таких стандартов занимается рабочая группа IETF под названием IPPM (IP Performance Metrics — метрики производительности IP-сетей). И хотя из названия группы видно, что ее стандарты ориентированы на характеристики именно IP-пакетов, эти стандарты носят достаточно общий характер, так что за исключением некоторых деталей могут применяться как основа для описания характеристик любых других протоколов (что и происходит на практике). Каждый стандарт имеет однотипную структуру. Сначала характеристика описывается как случайная величина, то есть дается определение ее единичного значения, которое является также значением ее единичного измерения. Затем дается описание того, что понимается под последовательностью замеров, то есть дается описание того, как правильно получить выборку значений характеристики. И наконец, приводятся рекомендуемые статистические оценки, которыми следует пользоваться при обработке полученной выборки значений. Обычно стандарты группы IPPM оставляют значительную свободу в выборе той или иной статистической оценки, рекомендуя несколько возможных оценок, например среднее значение, квантиль и максимальное значение.

Характеристики задержек пакетов

В этом разделе мы более формально рассмотрим характеристики производительности сети, относящиеся к задержкам и потерям пакетов.

Односторонняя задержка пакетов (One-Way Delay Metric, OWD) входит в число стандартов IPPM и описана в RFC 2679 (http://www.ietf.org/rfc/rfc2679.txt).

Единичное значение этой метрики описывается как время передачи пакета определенного типа между некоторыми двумя узлами сети. Под определенным типом понимается пакет, который имеет определенный набор заранее заданных признаков; стандарт жестко не оговаривает эти признаки, но указывает, что ими могут быть, например, размер пакета, тип приложения, сгенерировавшего пакет, тип протокола транспортного уровня, который доставил пакет, а также некоторые другие. Смысл используемого набора признаков состоит

в том, чтобы выделить из общего потока пакетов, приходящего в узел назначения, те пакеты, характеристики которых интересуют специалиста, проводящего измерения.

Единичное значение односторонней задержки пакетов определяется как интервал времени между моментом помещения в исходящую линию связи *первого* бита пакета узлом-отправителем и моментом приема *последнего* бита пакета с входящей линии связи узла-получателя.

Так как в этом определении учитывается время буферизации пакета узлом-получателем, то задержка зависит от размера пакета, и для получения сопоставимых результатов желательно в определении типа пакетов задавать определенный размер пакета. RFC 2679 не поясняет, почему было выбрано определение задержки, зависящее от размера пакета, но можно предполагать, что это связано с удобством измерения времени прихода пакета, так как программно его можно измерить только после завершения записи всего пакета в буфер операционной системы. Да и понять, относится ли пакет к нужному типу, при получении только его первого бита также невозможно.

В том случае, если пакет не прибыл в узел назначения за некоторое достаточно большое время (точное значение оставлено разработчику системы измерений), то пакет считается утерянным, а его задержка неопределенной (ее можно полагать равной бесконечности).

Последовательность замеров рекомендуется выполнять в случайные моменты времени, подчиняющиеся распределению Пуассона. Такой порядок выбора времени замеров позволяет избежать возможной синхронизации измерений с любыми периодическими флюктуациями в поведении сети, так как такая синхронизации может существенно исказить наблюдаемую картину.

И, наконец, RFC 2679 рекомендует использовать следующие *статистические оценки* для одностороннего времени задержки:

- квантиль для некоторого процента, при этом само значение процента не оговаривается;
- среднее значение задержки;
- минимальное значение задержки (в выборке).

Квантили удобны для оценки задержек в тех случаях, когда процент потерь пакетов достаточно высок, так что вычисление среднего значения задержки вызывает определенные трудности (можно игнорировать потери пакетов, но тогда мы получим слишком заниженную оценку). Для вычисления квантиля потерянные пакеты можно рассматривать как пакеты, пришедшие с бесконечно большой задержкой, которая, естественно, больше значения квантиля.

ПРИМЕЧАНИЕ

В некоторых случаях желательно иметь более однозначные рекомендации для выбираемых статистических оценок. На помощь здесь может прийти документ IETF, который на момент написания этой книги имел статус проекта стандарта Интернета. В этом проекте, называемом «Метрики IP-производительности для пользователей» (http://www.ietf.org/internet-drafts/draft-ietf-ippm-reporting-03.txt), приводятся более определенные рекомендации для основных характеристик производительности сети; к тому же выбранные оценки интуитивно понятны для пользователя. Так, в качестве оценки односторонней задержки в этом документе рекомендуется использовать медиану выборки.

Время реакции сети представляет собой интегральную характеристику производительности сети с точки зрения пользователя. Именно эту характеристику имеет в виду пользователь, когда говорит: «Сегодня сеть работает медленно».

Время реакции сети определяется как интервал времени между отправкой запроса пользователя к какой-либо сетевой службе и получением ответа на этот запрос.

Время реакции сети можно представить в виде нескольких слагаемых, например (рис. 6.6): времени подготовки запросов на клиентском компьютере ($t_{клиент1}$), времени передачи запросов между клиентом и сервером через сеть ($t_{сеть}$), времени обработки запросов на сервере ($t_{сервер}$), времени передачи ответов от сервера клиенту через сеть (снова $t_{сеть}$) и времени обработки получаемых от сервера ответов на клиентском компьютере ($t_{клиент2}$).

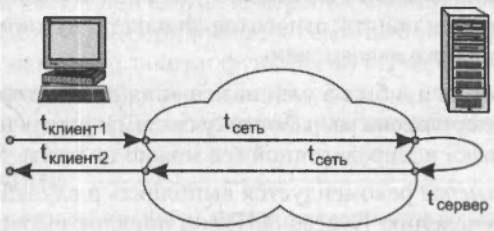


Рис. 6.6. Время реакции и время оборота

Время реакции сети характеризует сеть в целом, в том числе качество работы аппаратного и программного обеспечения серверов. Для того чтобы отдельно оценить транспортные возможности сети, используется другая характеристика — время оборота данных по сети.

Время оборота пакета (Round Trip Time, RTT) входит в число стандартов IPPM, описано в RFC 2681 (http://www.ietf.org/rfc/rfc2681.txt). Время оборота является составляющей времени реакции сети — это «чистое» время транспортировки данных от узла отправителя до узла назначения и обратно без учета времени, затраченного узлом назначения на подготовку ответа:

$$RTT = 2 \times t_{сеть}.$$

Единичное значение времени оборота определяется как интервал времени между отправкой первого бита пакета определенного типа узлом-отправителем узлу-получателю и получением последнего бита этого пакета узлом-отправителем после того, как пакет был получен узлом-получателем и отправлен обратно.

При этом узел-получатель должен отправить пакет узлу-отправителю как можно быстрее, чтобы не вносить искажения за счет времени обработки пакета.

RFC 2861 рекомендует ту же *последовательность замеров* времени оборота, что и для односторонней задержки, то есть случайные интервалы, подчиняющиеся распределению Пуассона.

RTT является удобной для измерений характеристикой, так как для ее получения не требуется синхронизация узла-отправителя и узла-получателя (узел-отправитель ставит временную отметку на отправляемый пакет, а затем по прибытии его от узла-получателя сравнивает эту отметку со своим текущим системным временем).

Однако информативность времени оборота меньше, чем односторонней задержки, так как информация о задержке в каждом направлении теряется, а это может затруднить поиск проблемного пути в сети.

Вариация задержки пакета (IP Packet Delay Variation, IPDV), которая входит в число стандартов IPPM, описана в RFC 3393 (http://www.ietf.org/rfc/rfc3393.txt).

Вариация задержки пакетов, которую также называют **джиттером** (jitter), очень важна для некоторых приложений. Так, при воспроизведении видеоклипа сама по себе задержка не очень существенна, например, если все пакеты задерживаются ровно на десять секунд, то качество воспроизведения не пострадает, а тот факт, что картинка появляется чуть позже, чем ее отослал сервер, пользователь даже не заметит (однако в интерактивных видеоприложениях, таких как видеоконференции, подобная задержка будет, конечно, уже ощутимо раздражать). А вот если задержки постоянно изменяются в пределах от нуля до 10 секунд, то качество воспроизведения клипа заметно ухудшится, для компенсации таких переменных задержек нужна предварительная буферизации поступающих пакетов в течение времени, превышающем вариацию задержки.

Единичное значение оценки вариации задержки определяется в RFC 3393 как разность односторонних задержек для пары пакетов заданного типа, полученных на интервале измерений *T*.

Как и для односторонней задержки, тип пакета может задаваться любыми признаками, однако для определенности измерений вариации задержки размеры обоих пакетов пары должны быть одинаковыми. Основной вопрос в этом определении — каким образом выбрать пару пакетов на интервале измерения *T*? Для ответа на этот вопрос в RFC 3393 вводится дополнительная функция — так называемая избирательная функция, которая и определяет правила выбора пары пакетов. Стандарт не определяет точное значение этой функции, он только говорит, что она должна существовать, и дает примеры возможных функций. Например, пары могут образовываться из всех последовательных пакетов, полученных на интервале; другим примером является выбор пакетов с определенными номерами в последовательности полученных пакетов, например пакетов с номерами 1, 5, 10, 15 и т. д. с интервалом 5.

Для оценки вариации задержки в соответствии с рекомендациями RFC 3393 требуется измерение задержек определенных пар пакетов. В то же время часто используется другой подход к определению вариации задержки, требующий только знания выборки односторонних задержек без их группировки в пары, отвечающие определенным условиям. Например, в уже упоминавшемся документе «Метрики IP-производительности для пользователей» в качестве оценки вариации задержки предлагается так называемый **разброс задержки** (delay spread). Разброс задержки определяется как разность между 75- и 25-процентными квантилями односторонней задержки. Таким образом, для того чтобы оценить вариацию задержки по этому определению, достаточно получить выборку значений односторонней задержки, а затем найти соответствующие квантили.

Характеристики скорости передачи

Скорость передачи данных (information rate) измеряется на каком-либо промежутке времени как частное от деления объема переданных данных за этот период на продолжительность периода. Таким образом, данная характеристика всегда является средней скоростью передачи данных.

Однако в зависимости от величины интервала, на котором измеряется скорость, для этой характеристики традиционно используется одно из двух наименований: средняя или пиковая скорость.

Средняя скорость передачи данных (Sustained Information Rate, SIR)[1] определяется на достаточно большом периоде времени. Это среднесрочная характеристика, период времени должен быть достаточным, чтобы можно было говорить об устойчивом поведении такой случайной величины, которой является скорость.

Должен быть оговорен период контроля этой величины, например 10 секунд. Это означает, что каждые 10 секунд вычисляется скорость информационного потока и сравнивается с требованием к этой величине. Если бы такие контрольные измерения не проводились, это лишило бы пользователя возможности предъявлять претензии поставщику в некоторых конфликтных ситуациях. Например, если поставщик в один из дней месяца вообще не будет передавать пользовательский трафик, а в остальные дни разрешит пользователю превышать оговоренный предел, то средняя скорость за месяц окажется в норме. В этой ситуации только регулярный контроль скорости поможет пользователю отстоять свои права.

Пиковая скорость передачи данных (Peak Information Rate, PIR) — это наибольшая скорость, которую разрешается достигать пользовательскому потоку в течение оговоренного небольшого периода времени T.

Этот период обычно называют **периодом пульсации**. Очевидно, что при передаче трафика можно говорить об этой величине только с некоторой степенью вероятности. Например, требование к этой характеристике может быть сформулировано так: «Скорость информации не должна превышать 2 Мбит/с на периоде времени 10 мс с вероятностью 0,95». Часто значение вероятности опускают, подразумевая близость ее к единице. Пиковая скорость является краткосрочной характеристикой. PIR позволяет оценить способность сети справляться с пиковыми нагрузками, характерными для пульсирующего трафика и приводящими к перегрузке. Если в SLA оговорены обе скорости (SIR и PIR), очевидно, что периоды пульсации должны сопровождаться периодами относительного «затишья», когда скорость падает ниже средней. В противном случае показатель средней скорости соблюдаться не будет.

Величина пульсации (обычно обозначаемая B) служит для оценки емкости буфера коммутатора, необходимого для хранения данных во время перегрузки. Величина пульсации равна общему объему данных, поступающих на коммутатор в течение разрешенного интервала T (периода пульсации) передачи данных с пиковой скоростью (PIR):

$$B = \mathrm{PIR} \times T.$$

Еще одной характеристикой скорости передачи является **коэффициент пульсации трафика** — это отношение максимальной скорости на каком-либо небольшом периоде времени к средней скорости трафика, измеренной на длительном периоде времени. Неопределенность временных периодов делает коэффициент пульсации *качественной* характеристикой трафика.

---

[1] Традиционно, для одной и той же характеристики может существовать несколько названий. Мы приводим только те из них, которые, по нашему мнению, наилучшим образом отражают их смысл.

Скорость передачи данных можно измерять между любыми двумя узлами, или точками, сети, например между клиентским компьютером и сервером, между входным и выходным портами маршрутизатора. Для анализа и настройки сети очень полезно знать данные о пропускной способности отдельных элементов сети.

Из-за последовательного характера передачи данных различными элементами сети общая пропускная способность любого составного пути в сети будет равна *минимальной* из пропускных способностей составляющих элементов маршрута.

Для повышения пропускной способности составного пути необходимо в первую очередь обратить внимание на самые медленные элементы, называемые **узкими местами** (bottleneck).

Надежность

Характеристики потерь пакетов

В качестве характеристики потерь пакетов используется **доля потерянных пакетов** (обозначим ее L), равная отношению количества потерянных пакетов (NL) к общему количеству переданных пакетов (N):

$$L = NL/N.$$

Может также использоваться аналогичная характеристика, оперирующая не количествами потерянных и переданных пакетов, а объемами данных, содержавшихся в этих пакетах.

Доступность и отказоустойчивость

Для описания надежности отдельных устройств служат такие показатели надежности, как **среднее время наработки на отказ**, **вероятность отказа**, **интенсивность отказов**. Однако эти показатели пригодны только для оценки надежности простых элементов и устройств, которые при отказе любого своего компонента переходят в неработоспособное состояние. Сложные системы, состоящие из многих компонентов, могут при отказе одного из компонентов сохранять свою работоспособность. В связи с этим для оценки надежности сложных систем применяется другой набор характеристик.

Доступность (availability) означает долю времени, в течение которого система или служба находится в работоспособном состоянии.

Доступность является долговременной статистической характеристикой, поэтому измеряется на большом промежутке времени, которым может быть день, месяц или год. Примером высокого уровня доступности является коммуникационное оборудование телефонных сетей, лучшие представители которого обладают так называемой доступностью «пять девяток». Это означает, что доступность равна 0,99999, что соответствует чуть более 5 минутам простоя в год. Оборудование и услуги передачи данных только стремятся к такому

рубежу, но рубеж трех девяток уже достигнут. Доступность услуги является универсальной характеристикой, которая важна как пользователям, так и поставщикам услуг.

Еще одной характеристикой надежности сложных систем является **отказоустойчивость** (fault tolerance). Под отказоустойчивостью понимается способность системы скрывать от пользователя отказ отдельных ее элементов.

Например, если коммутатор оснащен двумя коммутационными центрами, работающими параллельно, то отказ одного их них не приведет к полному останову коммутатора. Однако производительность коммутатора снизится, он будет обрабатывать пакеты вдвое медленней. В отказоустойчивой системе отказ одного из ее элементов приводит к некоторому снижению качества ее работы (**деградации**), а не к полному останову. В качестве еще одного примера можно назвать использование двух физических каналов для соединения коммутаторов. В нормальном режиме работы трафик передается по двум каналам со скоростью C Мбит/с, а при отказе одного из них трафик будут продолжать передаваться, но уже со скоростью $C/2$ Мбит/с. Однако из-за того, что во многих случаях количественно определить степени деградации системы или услуги достаточно сложно, отказоустойчивость чаще всего применяется как качественная характеристика.

Характеристики сети поставщика услуг

Рассмотрим основные характеристики, которыми оперирует поставщик услуг, оценивая эффективность своей сети. Эти характеристики часто являются качественными, то есть не могут быть выражены числами и соотношениями.

Расширяемость и масштабируемость

Термины «расширяемость» и «масштабируемость» иногда неверно используют как синонимы.

Расширяемость означает возможность сравнительно простого добавления отдельных компонентов сети (пользователей, компьютеров, приложений, служб), наращивания длины сегментов кабелей и замены существующей аппаратуры более мощной.

При этом принципиально важно, что простота расширения системы иногда может обеспечиваться в *определенных пределах*. Например, локальная сеть Ethernet, построенная на основе одного разделяемого сегмента коаксиального кабеля, обладает хорошей расширяемостью в том смысле, что позволяет легко подключать новые станции. Однако такая сеть имеет ограничение на число станций — оно не должно превышать 30–40. Хотя сеть допускает физическое подключение к сегменту и большего числа станций (до 100), при этом резко снижается производительность сети. Наличие такого ограничения и является признаком плохой масштабируемости системы при ее хорошей расширяемости.

Масштабируемость означает, что сеть позволяет наращивать количество узлов и протяженность связей в очень широких пределах, при этом производительность сети не снижается.

Для обеспечения масштабируемости сети приходится применять дополнительное коммуникационное оборудование и специальным образом структурировать сеть. Обычно масшта-

бируемое решение обладает многоуровневой иерархической структурой, которая позволяет добавлять элементы на каждом уровне иерархии без изменения главной идеи проекта.

Примером хорошо масштабируемой сети является Интернет, технология которого (TCP/IP) оказалась способной поддерживать сеть в масштабах земного шара. Организационная структура Интернета, которую мы рассмотрели в главе 5, образует несколько иерархических уровней: сети пользователей, сетей локальных поставщиков услуг и т. д. вверх по иерархии вплоть до сетей межнациональных поставщиков услуг. Технология TCP/IP, на которой построен Интернет, также позволяет строить иерархические сети. Основной протокол Интернета (IP) основан на двухуровневой модели: нижний уровень составляют отдельные сети (чаще всего сети корпоративных пользователей), а верхний уровень — это составная сеть, объединяющая эти сети. Стек TCP/IP поддерживает также концепцию автономной системы. В автономную систему входят все составные сети одного поставщика услуг, так что автономная система представляет собой более высокий уровень иерархии. Наличие автономных систем в Интернете позволяет упростить решение задачи нахождение оптимального маршрута — сначала ищется оптимальный маршрут между автономными системами, а затем каждая автономная система находит оптимальный маршрут внутри себя.

Не только сама сеть должна быть масштабируемой, но и устройства, работающие на магистрали сети, также должны обладать этим свойством, так как рост сети не должен приводить к необходимости постоянной смены оборудования. Поэтому магистральные коммутаторы и маршрутизаторы строятся обычно по модульному принципу, позволяя наращивать количество интерфейсов и производительность обработки пакетов в широких пределах.

Управляемость

Управляемость сети подразумевает возможность централизованно контролировать состояние основных элементов сети, выявлять и разрешать проблемы, возникающие при работе сети, анализировать производительность и планировать развитие сети. Управляемость предполагает наличие в сети некоторых автоматизированных средств администрирования, которые взаимодействуют с программным и аппаратным обеспечением сети с помощью коммуникационных протоколов.

В идеале средства администрирования сети представляют собой систему, осуществляющую *наблюдение и контроль* за каждым элементом сети — от простейших до самых сложных устройств, при этом сеть рассматривается как единое целое, а не как разрозненный набор отдельных устройств.

Хорошая система администрирования обеспечивает наблюдение за сетью и, обнаружив проблему, активизирует определенное действие, исправляет ситуацию и уведомляет администратора о том, что произошло и какие шаги предприняты. Одновременно с этим система администрирования должна *накапливать данные*, на основании которых можно планировать развитие сети. Наконец, система администрирования должна быть независима от производителя и обладать удобным интерфейсом, позволяющим выполнять все действия с одной консоли.

Решая тактические задачи, администраторы и технический персонал сталкиваются с ежедневными проблемами поддержания работоспособности сети. Эти задачи требуют быстрого решения, обслуживающий сеть персонал должен оперативно реагировать на сообщения о неисправностях, поступающих от пользователей или автоматических средств администрирования сети. Постепенно становятся заметными более общие проблемы производи-

тельности, конфигурирования сети, обработки сбоев и безопасности данных, требующие стратегического подхода, то есть *планирования* сети. Планирование, кроме того, подразумевает умение прогнозировать изменения в требованиях пользователей к сети, решение вопросов применения новых приложений, новых сетевых технологий и т. п.

Полезность системы администрирования особенно ярко проявляется в больших сетях: корпоративных или публичных глобальных. Без системы администрирования в таких сетях требуется присутствие квалифицированных специалистов по эксплуатации в каждом здании каждого города, где установлено оборудование сети, что в итоге приводит к необходимости содержания огромного штата обслуживающего персонала.

В настоящее время в области систем администрирования сетей накопилось много нерешенных проблем. Явно недостаточно действительно удобных, компактных и многопротокольных средств администрирования. Большинство существующих средств вовсе не управляют сетью, а всего лишь обеспечивают *наблюдение* за ее работой и *фиксацию* важных событий, например отказов устройств. Реже системы администрирования выполняют активные действия, ликвидирующие последствия нежелательного события или предотвращающие его.

Совместимость

Совместимость, или **интегрируемость**, сети означает, что сеть способна включать в себя самое разнообразное программное и аппаратное обеспечение, то есть в ней могут сосуществовать различные операционные системы, поддерживающие разные стеки коммуникационных протоколов, а также аппаратные средства и приложения от разных производителей. Сеть, состоящая из разнотипных элементов, называется неоднородной, или гетерогенной, а если гетерогенная сеть работает без проблем, то она является интегрированной. Основной путь построения интегрированных сетей — использование модулей, выполненных в соответствии с открытыми стандартами и спецификациями.

Выводы

Главным требованием, предъявляемым к компьютерной сети, является обеспечение высокого качества предоставляемых сетью услуг. При широком понимании в понятие «качество обслуживания» включают все возможные характеристики услуг и сети, желательные для пользователя. Наиболее важные формализованные характеристики сети относятся к ее производительности и надежности.

Производительность сети оценивается с помощью статистических характеристик двух типов: характеристик скорости передачи информации и характеристик задержек передачи пакетов. В первую группу входят средняя скорость и максимальная скорость на периоде пульсации, а также длительность этого периода. Во вторую группу входят: средняя величина задержки, средняя вариация задержки (джиттер), коэффициент вариации, а также максимальные значения задержки и вариации задержки.

Для оценки надежности сетей применяются различные характеристики, в том числе: доля потерь пакетов, коэффициент доступности, означающий долю времени, в течение которого система может быть использована, отказоустойчивость — способность системы работать в условиях отказа некоторых ее элементов.

Надежность транспортных услуг сети обеспечивается надежностью ее компонентов (каналов и коммуникационного оборудования), наличием альтернативных маршрутов, а также повторной передачей потерянных или искаженных пакетов.

Особую важность для поставщика услуг представляют такие качественные характеристики сети, как ее масштабируемость, расширяемость и управляемость.

Вопросы и задания

1. Могут ли различаться краткосрочные и долгосрочные значения одной и той же характеристики, например средней скорости потока?
2. Что желательно оговорить в разделе соглашения SLA, относящегося ко времени задержек пакетов?
3. Какие составляющие задержки пакета являются фиксированными для пакета фиксированной длины?
4. Какая составляющая задержки пакета зависит от длины пакета?
5. Каким образом передает пакеты идеальная сеть? Какие из вариантов ответов вы считаете верными:

   а) не потеряв ни один из пакетов (и не исказив информацию ни в одном из них);

   б) в том порядке, в котором они были отправлены;

   в) с одной и той же и минимально возможной задержкой, определяемой временем распространения сигнала по среде линий связи.
6. Найдите медиану и среднее значение следующей выборки значений задержки пакетов (в мс):

   10, 12, 15, 17, 18, 20, 10 000.
7. Как вы думаете, какая из оценок задержек, медиана или среднее значение, лучше характеризует задержки в сети, представленные выборкой из задания 6?
8. Найдите 85-процентный квантиль для выборки значений задержки пакетов из задания 6.
9. Чем метод активных измерений отличается от схемы пассивных измерений?
10. Зависит ли единичное значения односторонней задержки пакета, определенное в RFC 2679, от размера пакета?
11. В чем заключаются положительные и отрицательные стороны использования времени оборота в качестве характеристики задержек пакетов в сети?
12. Каким образом можно компенсировать вариацию задержки?
13. Что формирует избирательная функция?
14. Что из приведенного ниже может учитывать избирательная функция:

   а) время поступления пакетов;

   б) номера пакетов в выборке;

   в) разницу задержек пакетов.
15. Приведите пример выборки задержки пакетов на некотором интервале времени, на котором средняя скорость потока пакетов, поступающих на узел-получатель, отличается от средней скорости пакетов, генерируемых узлом-отправителем.
16. Может ли трафик передаваться с большими задержками, но без джиттера?
17. Объясните разницу между масштабируемостью и расширяемостью.

ГЛАВА 7 Методы обеспечения качества обслуживания

Методы обеспечения качества обслуживания (QoS) занимают сегодня важное место в арсенале технологий сетей с коммутацией пакетов, так как они обеспечивают устойчивую работу современных мультимедийных приложений, таких как IP-телефония, видео- и радиовещание, интерактивное дистанционное обучение и т. п. Методы QoS направлены на улучшение характеристик производительности и надежности сети, рассмотренных в предыдущей главе; эти методы позволяют уменьшить задержки, вариации задержек, а также потери пакетов в периоды перегрузки сети, создавая тем самым необходимые условия для удовлетворительного обслуживания сетью трафика приложений.

Методы обеспечения качества обслуживания направлены на компенсацию негативных последствий временны́х перегрузок, возникающих в сетях с коммутацией пакетов. В этих методах используются различные алгоритмы управления очередями, резервирования и обратной связи, позволяющие снизить негативное влияние очередей до приемлемого для пользователей уровня.

Обзор методов обеспечения качества обслуживания

Очереди являются неотъемлемым атрибутом сетей с коммутацией пакетов. Сам принцип работы таких сетей подразумевает наличие буфера у каждых входного и выходного интерфейсов коммутатора пакетов. Буферизация пакетов во время перегрузок представляет собой основной механизм поддержания пульсирующего трафика, обеспечивающий высокую производительность сетей этого типа. Как вы знаете, в сетях с другим типом коммутации, а именно в сетях с коммутацией каналов, промежуточная буферизация данных не поддерживается. В то же время очереди означают неопределенную задержку при передаче пакетов через сеть, а в некоторых случаях и потери пакетов из-за переполнения буфера коммутатора или маршрутизатора, отведенного под очередь. Задержки и потери пакетов — это главный источник проблем для чувствительного к задержкам трафика. Так как сегодня операторы пакетных сетей очень заинтересованы в передаче пульсирующего трафика, им необходимы средства достижения компромисса между требованиями предельной загрузки своей сети и качеством обслуживания одновременно всех типов трафика.

Существует два подхода к определению того, какие характеристики производительности и надежности следует отнести к **характеристикам качества обслуживания**, то есть к тем характеристикам, которые могут быть улучшены с помощью методов QoS.

В одном случае, под характеристиками QoS понимается только три характеристики:

- односторонняя задержка пакетов;
- вариация задержек пакетов;
- потери пакетов.

Другой подход заключается в расширенном толковании характеристик QoS, когда характеристики скорости потока, такие как средняя скорость, пиковая скорость и пульсация, также относят к характеристикам QoS.

В *методах обеспечения качества обслуживания* используются различные механизмы, направленные на снижение негативных последствий пребывания пакетов в очередях с сохранением в то же время положительной роли очередей. Набор механизмов достаточно широк, и в этой главе они рассматриваются достаточно подробно. Большинство из них учитывает и использует в своей работе факт существования в сети трафика различного типа в том отношении, что каждый тип трафика предъявляет различные требования к характеристикам производительности и надежности сети. Например, трафик просмотра веб-страниц мало чувствителен к задержкам пакетов и не требует гарантированной пропускной способности сети, зато чувствителен к потерям пакетов; в то же время как голосовой трафик очень чувствителен к задержкам пакетов, требует гарантированной пропускной способности сети, но может «терпеть» потерю небольшого процента пакетов без значительного ущерба для качества (впрочем, последнее свойство во многом зависит от используемого метода кодирования голосового сигнала).

Добиться одновременного соблюдения *всех* характеристик QoS для *всех* видов трафика весьма сложно. Одним из наиболее значимых факторов, влияющих на характеристики качества обслуживания, является уровень загрузки сети трафиком, то есть уровень использования пропускной способности линий связи сети.

Если этот уровень постоянно достаточно низок, то трафик всех приложений обслуживается с высоким качеством большую часть времени (хотя кратковременные перегрузки сети,

приводящие к задержкам и потерям пакетов, все равно возможны, но они случаются очень редко). Такое состояние сети называется «недогруженным» или же используется термин **сеть с избыточной пропускной способностью** (англоязычный термин overprovisioning). Постоянно поддерживать все части сети в недогруженном состоянии достаточно дорого и сложно, но для наиболее ответственной части сети, такой как магистраль, этот подход применяется, и связан он с постоянным слежением за уровнем загрузки каналов магистрали и периодическим увеличением их пропускной способности по мере приближения загрузки к критическому уровню.

Методы QoS основаны на другом подходе, а именно тонком перераспределении имеющейся пропускной способности между трафиком различного типа в соответствии с требованиями приложений. Очевидно, что эти методы усложняют сетевые устройства, так как означают необходимость знать требования всех классов трафика, уметь их классифицировать и распределять пропускную способность сети между ними. Последнее свойство обычно достигается за счет использования нескольких очередей пакетов для каждого выходного интерфейса коммуникационного устройства вместо одной очереди; при этом в очередях применяют различные алгоритмы обслуживания пакетов, чем и достигается дифференцированное обслуживание трафика различных классов. Поэтому методы QoS часто ассоциируются с *техникой управления очередями*.

Помимо собственно техники организации очередей, к методам QoS относят методы контроля параметров потока трафика, так как для гарантированно качественного обслуживания нужно быть уверенными, что обслуживаемые потоки соответствуют определенному профилю. Эта группа методов QoS получила название *методов кондиционирования трафика*.

Особое место занимают *методы обратной связи*, которые предназначены для уведомления источника трафика о перегрузке сети. Эти методы рассчитаны на то, что при получении уведомления источник снизит скорость выдачи пакетов в сеть и тем самым ликвидирует причину перегрузки.

Механизмы QoS можно применять по-разному. В том случае, когда они применяются к отдельным узлам без учета реальных маршрутов следования потоков трафика через сеть[1], условия обслуживания трафика этими узлами улучшаются, но гарантий того, что поток будет обслужен с заданным уровнем качества, такой подход не дает. Гарантии можно обеспечить, если применять методы QoS системно, *резервируя ресурсы сети* для потока на всем протяжении его маршрута, другими словами, «из конца в конец».

К методам QoS тесно примыкают *методы инжиниринга трафика*. Согласно методам инжиниринга трафика маршруты передачи данных управляются таким образом, чтобы обеспечить сбалансированную загрузку всех ресурсов сети и исключить за счет этого перегрузку коммуникационных устройств и образование длинных очередей. В отличие от методов QoS в методах инжиниринга трафика не прибегают к организации очередей с различными алгоритмами обслуживания на сетевых устройствах. В то же время в методах QoS в их традиционном понимании не используют такой мощный рычаг воздействия на рациональное распределение пропускной способности, как изменение маршрутов трафика в зависимости от фактической загрузки линий связи, что позволяет легко отделить методы QoS от методов инжиниринга трафика.

В следующей группе методов борьба с перегрузками ведется путем *снижения постоянной нагрузки на сеть*. То есть в этих методах проблема рассматривается с другой стороны:

---

[1] Так называемое «поузловое» (per hop) применение.

если пропускной способности сети недостаточно для качественной передачи трафика приложений, то нельзя ли уменьшить объем самого трафика? Наиболее очевидным способом снижения объема трафика является его *компрессия*; существуют и другие способы, приводящие к тому же результату, например размещение источника данных ближе к его потребителю (*кэширование данных*).

Приложения и качество обслуживания

К настоящему времени проделана большая работа по классификации трафика приложений. В качестве основных критериев классификации были приняты три характеристики трафика:

- относительная предсказуемость скорости передачи данных;
- чувствительность трафика к задержкам пакетов;
- чувствительность трафика к потерям и искажениям пакетов.

Предсказуемость скорости передачи данных

В отношении предсказуемости скорости передачи данных приложения делятся на два больших класса: приложения с потоковым трафиком и приложения с пульсирующим трафиком.

Приложения с потоковым трафиком (stream) порождают равномерный поток данных, который поступает в сеть с **постоянной битовой скоростью** (Constant Bit Rate, CBR). В случае коммутации пакетов трафик таких приложений представляет собой последовательность пакетов одинакового размера (равного B бит), следующих друг за другом через один и тот же интервал времени T (рис. 7.1).

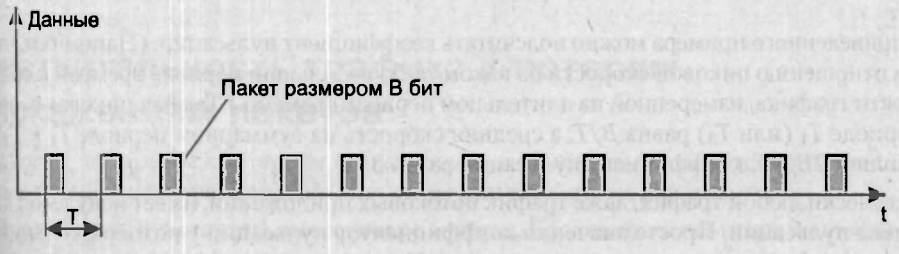


Рис. 7.1. Потоковый трафик

Постоянная битовая скорость потокового трафика (CBR) может быть вычислена путем усреднения на одном периоде:

$$CBR = B/T \text{ бит/с.}$$

В общем случае, постоянная битовая скорость потокового трафика меньше номинальной максимальной битовой скорости протокола, с помощью которого передаются данные, так как между пакетами существуют паузы. Как будет показано в главе 12, максимальная скорость передачи данных с помощью протокола Ethernet составляет 9,76 Мбит/с (для

кадров максимальной длины), что меньше номинальной скорости этого протокола, равной 10 Мбит/с.

Приложения с пульсирующим трафиком (burst) отличаются высокой степенью непредсказуемости, в этих приложениях периоды молчания сменяются пульсацией, в течение которой пакеты «плотно» следуют друг за другом. В результате трафик характеризуется **переменной битовой скоростью** (Variable Bit Rate, VBR), что иллюстрирует рис. 7.2. Так, при работе приложений файлового сервиса интенсивность трафика, генерируемого приложением, может падать до нуля, когда файлы не передаются, и повышаться до максимально доступной, ограниченной только возможностями сети, когда файловый сервер передает файл.

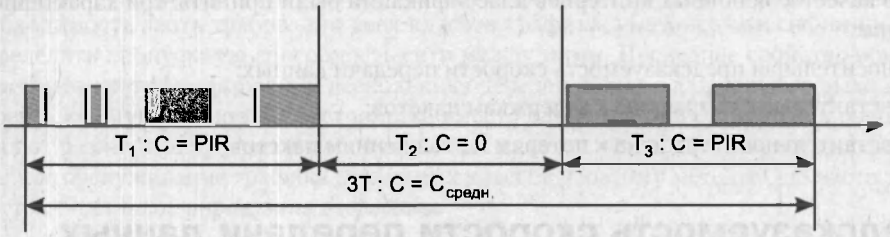


Рис. 7.2. Пульсирующий трафик

На рисунке показано три периода измерений T_1, T_2 и T_3. Для упрощения расчетов принято, что пиковые скорости на первом и третьем периодах равны между собой и равны PIR, а все три периода имеют одинаковую длительность T. Учитывая это, можно вычислить величину пульсации B, которая равна количеству битов, переданных на периоде пульсации:

$$B = \text{PIR} \times T.$$

Таким образом, величина пульсации для периодов T_1 и T_3 равна B, а на периоде T_2 — нулю.

Для приведенного примера можно подсчитать коэффициент пульсации. (Напомним, что он равен отношению пиковой скорости на каком-либо небольшом периоде времени к средней скорости трафика, измеренной на длительном периоде времени.) Так как пиковая скорость на периоде T_1 (или T_3) равна B/T, а средняя скорость на суммарном периоде $T_1 + T_2 + T_3$ составляет $2B/3T$, коэффициент пульсации равен 3/2.

Практически любой трафик, даже трафик потоковых приложений, имеет ненулевой коэффициент пульсации. Просто значения коэффициентов пульсации у потокового и пульсирующего трафиков существенно различаются. У приложений с пульсирующим трафиком он обычно находится в пределах от 2 до 100, а у потоковых приложений он близок к 1. В локальных сетях коэффициент пульсации обычно выше, чем в глобальных, поскольку на магистралях глобальных сетей трафик представляет собой сумму трафиков многих источников, что по закону больших чисел приводит к сглаживанию результирующего трафика.

Чувствительность трафика к задержкам пакетов

Еще один критерий классификации приложений по типу трафика — их чувствительность к задержкам пакетов и их вариациям. Далее перечислены основные типы приложений в порядке повышения чувствительности к задержкам пакетов.

- **Асинхронные приложения.** Практически не имеют ограничений на время задержки (эластичный трафик). Пример такого приложения — электронная почта.
- **Интерактивные приложения.** Задержки могут быть замечены пользователями, но они не сказываются негативно на функциональности приложений. Пример — текстовый редактор, работающий с удаленным файлом.
- **Изохронные приложения.** Имеется порог чувствительности к вариациям задержек, при превышении которого резко снижается функциональность приложения. Пример — передача голоса, когда при превышении порога вариации задержек в 100–150 мс резко снижается качество воспроизводимого голоса.
- **Сверхчувствительные к задержкам приложения.** Задержка доставки данных сводит функциональность приложения к нулю. Пример — приложения, управляющие техническим объектом в реальном времени. При запаздывании управляющего сигнала на объекте может произойти авария.

Вообще говоря, интерактивность приложения всегда повышает его чувствительность к задержкам. Например, широковещательная рассылка аудиоинформации может выдерживать значительные задержки передачи пакетов (оставаясь чувствительным к вариациям задержек), а интерактивный телефонный или телевизионный разговор их не терпит, что хорошо заметно при трансляции разговора через спутник. Длительные паузы в разговоре вводят собеседников в заблуждение, часто они теряют терпение и начинают очередную фразу одновременно.

Наряду с приведенной классификацией, тонко дифференцирующей чувствительность приложений к задержкам и их вариациям, существует и более грубое деление приложений по этому же признаку на два класса: асинхронные и синхронные. К *асинхронным* относят те приложения, которые нечувствительны к задержкам передачи данных в очень широком диапазоне, вплоть до нескольких секунд, а все остальные приложения, на функциональность которых задержки влияют существенно, относят к *синхронным* приложениям.

Интерактивные приложения могут относиться как к асинхронным (например, текстовый редактор), так и к синхронным (например, видеоконференция).

Чувствительность трафика к потерям и искажениям пакетов

И, наконец, последним критерием классификации приложений является их чувствительность к потерям пакетов. Здесь обычно делят приложения на две группы.

- **Приложения, чувствительные к потере данных.** Практически все приложения, передающие алфавитно-цифровые данные (к которым относятся текстовые документы, коды программ, числовые массивы и т. п.), обладают высокой чувствительностью к потере отдельных, даже небольших, фрагментов данных. Такие потери часто ведут к полному обесцениванию остальной успешно принятой информации. Например, отсутствие хотя бы одного байта в коде программы делает ее совершенно неработоспособной. Все традиционные сетевые приложения (файловый сервис, сервис баз данных, электронная почта и т. д.) относятся к этому типу приложений.
- **Приложения, устойчивые к потере данных.** К этому типу относятся многие приложения, передающие трафик с информацией об инерционных физических процессах. Устойчивость к потерям объясняется тем, что небольшое количество отсутствующих данных можно определить на основе принятых. Так, при потере одного пакета, несущего

несколько последовательных замеров голоса, отсутствующие замеры при воспроизведении голоса могут быть заменены аппроксимацией на основе соседних значений. К такому типу относится большая часть приложений, работающих с мультимедийным трафиком (аудио- и видеоприложения). Однако устойчивость к потерям имеет свои пределы, поэтому процент потерянных пакетов не может быть большим (например, не более 1 %). Можно отметить также, что не любой мультимедийный трафик столь устойчив к потерям данных, так, компрессированный голос и видеоизображение очень чувствительны к потерям, поэтому относятся к первому типу приложений.

Классы приложений

Вообще говоря, между значениями трех характеристик качества обслуживания (относительная предсказуемость скорости передачи данных; чувствительность трафика к задержкам пакетов; чувствительность трафика к потерям и искажениям пакетов) нет строгой взаимосвязи. То есть приложение с равномерным потоком может быть как асинхронным, так и синхронным, а, например, синхронное приложение может быть как чувствительным, так и нечувствительным к потерям пакетов. Однако практика показывает, что из всего многообразия возможных сочетаний значений этих трех характеристик есть несколько таких, которые охватывают большую часть используемых сегодня приложений.

Например, следующее сочетание характеристик приложения «порождаемый трафик — равномерный поток, приложение изохронное, устойчивое к потерям» соответствует таким популярным приложениям, как IP-телефония, поддержка видеоконференций, аудиовещание через Интернет. Устойчивых сочетаний характеристик, описывающих определенный класс приложений, существует не так уж много. Так, при стандартизации технологии ATM, которая изначально разрабатывалась для поддержания различных типов трафика, были определены 4 класса трафика (и соответствующих приложений): *A*, *B*, *C* и *D*. Для каждого класса рекомендуется использовать собственный набор характеристик QoS. Кроме того, для всех приложений, не включенных ни в один из этих классов, был определен класс *X*, в котором сочетание характеристик приложения может быть произвольным.

Классификация ATM, являясь на сегодня наиболее детальной и обобщенной, не требует для своего понимания знания технологии ATM, поэтому мы можем рассмотреть ее уже сейчас (табл. 7.1).

Таблица 7.1. Классы трафика

Класс трафика	Характеристики
A	Постоянная битовая скорость, чувствительность к задержкам, передача с установлением соединения (например, голосовой трафик, трафик телевизионного изображения). Параметры QoS: пиковая скорость передачи данных, задержка, джиттер
B	Переменная битовая скорость, чувствительность к задержкам, передача с установлением соединения (например, компрессированный голос, компрессированное видеоизображение). Параметры QoS: пиковая скорость передачи данных, пульсация, средняя скорость передачи данных, задержка, джиттер
C	Переменная битовая скорость, эластичность, передача с установлением соединения (например, трафик компьютерных сетей, в которых конечные узлы работают по протоколам с установлением соединений — frame relay, X.25, TCP). Параметры QoS: пиковая скорость передачи данных, пульсация, средняя скорость передачи данных

Класс трафика	Характеристики
D	Переменная битовая скорость, эластичность, передача без установления соединения (например, трафик компьютерных сетей, в которых конечные узлы работают по протоколам без установления соединений — IP/UDP, Ethernet). Параметры QoS не определены
X	Тип трафика и его параметры определяются пользователем

Анализ очередей

Определить основные характеристики QoS и сформулировать требования к ним — значит, наполовину решить задачу. Пользователь формулирует свои требования к качеству обслуживания в виде некоторых предельных значений характеристик QoS, которые не должны быть превышены, например он может указать, что предельное значение вариации задержки пакетов не должно превышать 50 мс с вероятностью 0,99.

Но как заставить сеть справиться с поставленной задачей? Какие меры нужно предпринять, чтобы вариации задержек действительно не превысили эту величину? И как гарантировать пользователю, что средняя скорость передачи его потока через сеть будет соответствовать средней скорости входящего в сеть потока?

Для понимания механизмов поддержки QoS полезно исследовать процесс образования очередей в сетевых устройствах и понять наиболее существенные факторы, влияющие на длину очереди.

Модель M/M/1

Существует ветвь прикладной математики, предметом которой являются процессы образования очередей. Эта дисциплина так и называется — **теория очередей**. Мы не будем углубляться в математические основы этой теории, приведем только некоторые ее выводы, существенные для рассматриваемой нами проблемы QoS.

На рис. 7.3 показана наиболее простая модель очереди, известная под названием M/M/1[1].

Рис. 7.3. Модель M/M/1

Основными элементами модели являются:

- входной поток абстрактных заявок на обслуживание;
- буфер;

[1] Здесь 1 означает, что моделируется одно обслуживающее устройство, первая буква *M* обозначает тип распределения интервалов поступления заявок (марковское распределение), вторая — тип распределения значений времени обслуживания (тоже марковское).

- обслуживающее устройство;
- выходной поток обслуженных заявок.

Заявки поступают на вход буфера в случайные моменты времени. Если в момент поступления заявки буфер пуст и обслуживающее устройство свободно, то заявка сразу же передается в это устройство для обслуживания. Обслуживание также длиться случайное время.

Если в момент поступления заявки буфер пуст, но обслуживающее устройство занято обслуживанием ранее поступившей заявки, то заявка ожидает его завершения в буфере. Как только обслуживающее устройство завершает обслуживание очередной заявки, она передается на выход, а прибор выбирает из буфера следующую заявку (если буфер не пуст). Выходящие из обслуживающего устройства заявки образуют выходной поток. Буфер считается бесконечным, то есть заявки никогда не теряются из-за того, что исчерпана емкость буфера.

Если прибывшая заявка застает буфер не пустым, то она становится в очередь и ожидает обслуживания. Заявки выбираются из очереди в порядке поступления, то есть соблюдается дисциплина обслуживания **первым пришел — первым обслужен** (First-In, First-Out, FIFO).

Теория очередей позволяет оценить для этой модели среднюю длину очереди и среднее время ожидания заявки в очереди в зависимости от характеристик входного потока и времени обслуживания.

Будем считать, что среднее время между поступлениями заявок известно и равно T. Это значит, что интенсивность поступления заявок, которая традиционно обозначается в теории очередей символом λ, равна

$$\lambda = 1/T \text{ заявок в секунду.}$$

Случайный процесс поступления заявок описывается в этой модели функцией распределения интервалов между поступлениями заявок. Чтобы упростить получение компактных аналитических результатов, обычно считают, что эти интервалы описываются так называемым **марковским** распределением (другое название — **пуассоновское**), плотность которого показана на рис. 7.4. Из рисунка видно, что входной поток является существенно пульсирующим, так как есть ненулевая вероятность того, что интервал между заявками будет очень небольшим, близким к нулю, а также того, что он будет очень большим. Среднее отклонение интервалов также равно T.

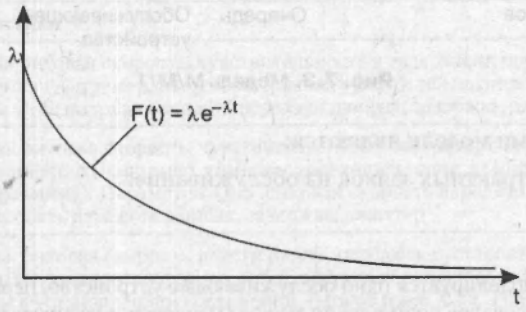


Рис. 7.4. Плотность распределения входного потока

Будем также считать, что среднее время обслуживания заявки равно b. Это означает, что обслуживающий прибор способен продвигать заявки на выход с интенсивностью $1/b = \mu$. Опять же для получения аналитического результата считают, что время обслуживания — это случайная величина с пуассоновской плотностью распределения.

Принятие таких предположений дает простой результат для среднего времени ожидания заявки в очереди, которое мы обозначим через w:

$$w = \rho \frac{b}{1-\rho}. \qquad (1)$$

Здесь через ρ обозначено отношение λ/μ.

Параметр ρ называют **коэффициентом использования** (utilization) обслуживающего прибора. Для любого периода времени этот показатель равен отношению времени занятости обслуживающего прибора к величине этого периода.

Зависимость среднего времени ожидания заявки w от ρ иллюстрирует рис. 7.5. Как видно из поведения кривой, параметр ρ играет ключевую роль в образовании очереди. Если значение ρ близко к нулю, то среднее время ожидания тоже очень близко к нулю. А это означает, что заявки почти никогда не ожидают обслуживания в буфере (в момент их прихода он оказывается пустым), а сразу попадают в обслуживающее устройство. И наоборот, если ρ приближается к 1, то время ожидания растет очень быстро и нелинейно (и в пределе равно бесконечности). Такое поведение очереди интуитивно понятно, ведь ρ — это отношение средней интенсивности входного потока к средней интенсивности его обслуживания. Чем ближе средние значения интервалов между пакетами к среднему времени обслуживания, тем сложнее обслуживающему устройству справляться с нагрузкой.

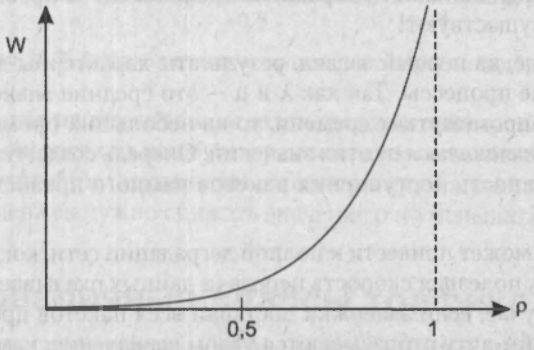


Рис. 7.5. Зависимость среднего времени ожидания заявки от коэффициента использования ресурса

С помощью модели M/M/1 можно приближенно моделировать сеть с коммутацией пакетов (рис. 7.6).

Так, входной поток пакетов, поступающих на вход интерфейса коммутатора (будем здесь использовать этот термин как обобщенное название устройства коммутации пакетов), представлен в модели потоком заявок, а буфер модели соответствует буферу интерфейса коммутатора. Среднее время обслуживания заявки соответствует среднему времени продвижения пакета процессором коммутатора из входного буфера в выходной канал.

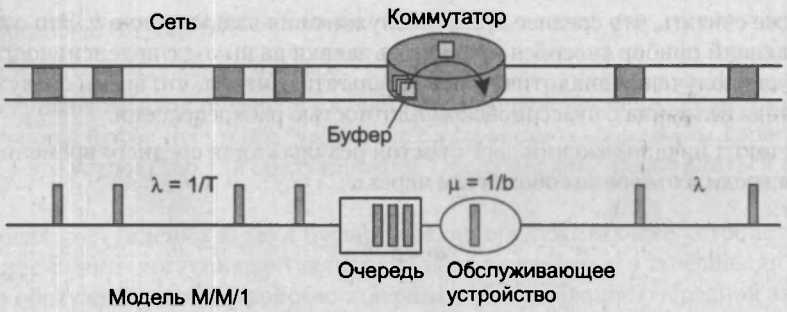


Рис. 7.6. Выходной интерфейс как разделяемый ресурс коммутатора

Понятно, что приведенная модель очень упрощенно описывает процессы, происходящие в коммутаторе. Тем не менее она очень полезна для понимания основных факторов, влияющих на величину очереди.

Сетевые инженеры хорошо знакомы с графиком, представленным на рис. 7.5. Они интерпретируют этот график как зависимость задержек в сети от ее загрузки. Параметр ρ модели соответствует коэффициенту использования сетевого ресурса, который участвует в передаче трафика, то есть выходного интерфейса коммутатора.

В приведенном графике есть и нечто неожиданное. Трудно представить, что обслуживающее устройство (сетевой ресурс) практически перестает справляться со своими обязанностями, когда его коэффициент использования приближается к 1. Ведь в этом случае нагрузка не превышает его возможностей, а только приближается к этому пределу. Интуитивно не очень понятна также причина существования очередей при значениях ρ в окрестностях 0,5. Интенсивность обработки трафика вдвое превышает интенсивность нагрузки, а очереди существуют!

Такие парадоксальные, на первый взгляд, результаты характерны для систем, в которых протекают случайные процессы. Так как λ и μ — это средние значения интенсивностей потоков на больших промежутках времени, то на небольших промежутках времени они могут *существенно отклоняться* от этих значений. Очередь создается на тех промежутках, на которых интенсивность поступления пакетов намного превосходит интенсивность обслуживания.

Перегрузка ресурсов может привести к полной деградации сети, когда, несмотря на то что сеть передает пакеты, полезная скорость передачи данных оказывается равной нулю. Это происходит в том случае, если задержки доставки всех пакетов превосходят некоторый порог, и пакеты по тайм-ауту отбрасываются узлом назначения, как устаревшие. Если же протоколы, работающие в сети, используют надежные процедуры передачи данных на основе квитирования и повторной передачи утерянных пакетов, то процесс перегрузки будет нарастать лавинообразно.

Существует еще один важный параметр, оказывающий непосредственное влияние на образование очередей в сетях с коммутацией пакетов. Этим параметром является вариация интервалов входного потока пакетов, то есть пульсация входного трафика. Мы анализировали поведение модели теории очередей в предположении, что входной поток описывается пуассоновским распределением, которое имеет довольно большое стандартное отклонение

вариации (напомним, что средняя вариация его равна T при среднем значении интервала T, а коэффициент вариации равен 1). А что будет, если вариация интервалов входного потока будет меньше? Или входной поток окажется сверхпульсирующим?

К сожалению, модели теории очередей не дают для этих случаев простых аналитических зависимостей, подобных формуле (1). Поэтому для получения результатов приходится применять методы имитационного моделирования сетей или проводить измерения в реальной сети.

На рисунке 7.7 показано семейство зависимостей w от ρ, полученных для разных значений коэффициента вариации CV входного потока. Имитационная модель учитывает фиксированную задержку в сети. Одна из кривых, у которой CV = 1, соответствует пуассоновскому входному потоку. Из рисунка видно, что чем меньше пульсирует входной поток (CV приближается к нулю), тем меньше проявляется эффект лавинообразного образования очереди при приближении коэффициента загрузки ресурса к 1. И наоборот, чем больше CV, тем раньше (при меньших значениях ρ) начинает этот эффект проявляться.

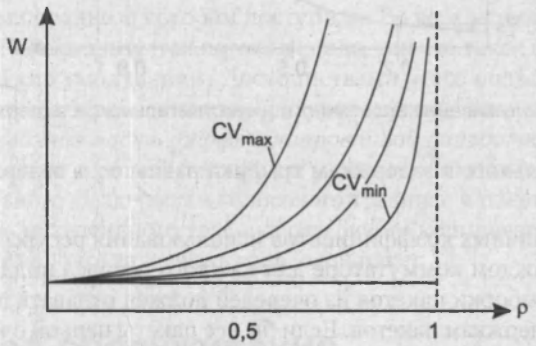


Рис. 7.7. Влияние степени пульсации потока на задержки

Из поведения графиков на рисунке можно сделать два вывода: во-первых, для оценки значений задержек в очередях на коммутаторах сети недостаточно информации о коэффициенте загрузки ρ, необходимо также знать параметры пульсации трафика; во-вторых, для снижения уровня задержек нужно снижать значение ρ и уменьшать пульсацию трафика.

Очереди и различные классы трафика

Посмотрим, как можно применить наши знания о зависимости поведения очередей от коэффициента загрузки для реализации основной идеи методов QoS, а именно дифференцированного обслуживания классов трафика с различными требованиями к характеристикам производительности и надежности сети. Чтобы проще было в этом разобраться, будем пока делить все потоки на два класса — чувствительный к задержкам (трафик реального времени, например голосовой) и эластичный, допускающий большие задержки, но чувствительный к потерям данных.

Мы знаем, что если обеспечить для чувствительного к задержкам трафика коэффициент загрузки каждого ресурса не более 0,2, то, очевидно, задержки в каждой очереди будут

небольшими и, скорее всего, приемлемыми для многих типов приложений этого класса. Для эластичного трафика, слабо чувствительного к задержкам, можно допустить более высокий коэффициент загрузки, но не более 0,9. Для того чтобы пакеты этого класса не терялись, нужно предусмотреть для них буферную память, достаточную для хранения всех пакетов периода пульсации. Эффект от такого распределения загрузки ресурса иллюстрирует рис. 7.8.

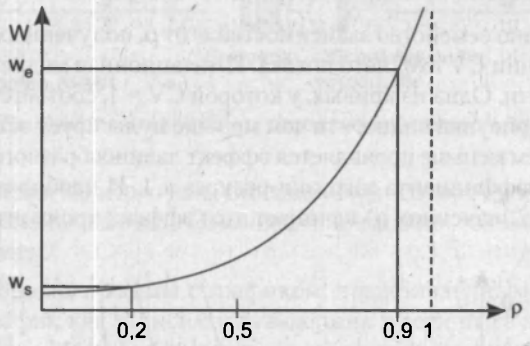


Рис. 7.8. Обслуживание эластичного и чувствительного к задержкам трафика

Задержки чувствительного к задержкам трафика равны w_s, а задержки эластичного трафика — w_e.

Чтобы добиться различных коэффициентов использования ресурсов для разных классов трафика, нужно в каждом коммутаторе для каждого ресурса поддерживать две разные очереди. Алгоритм выборки пакетов из очередей должен отдавать предпочтение очереди чувствительных к задержкам пакетов. Если бы все пакеты первой очереди обслуживались приоритетно, а пакеты второй очереди — только тогда, когда первая очередь пуста, то для трафика первой очереди трафик второй очереди фактически перестал бы существовать. Поэтому если отношение средней интенсивности приоритетного трафика λ_1 к производительности ресурса μ равно 0,2, то и коэффициент загрузки для него равен 0,2. А вот для эластичного трафика, пакеты которого всегда ждут обслуживания приоритетных пакетов, коэффициент загрузки подсчитывается по-другому. Если средняя интенсивность эластичного трафика равна λ_2, то для него ресурс будет загружен на $(\lambda_1 + \lambda_2)/\mu$. Так что если мы хотим, чтобы для эластичного трафика коэффициент загрузки составлял 0,9, то его интенсивность должна находиться из соотношения $\lambda_2/\mu = 0,7$.

Основная идея, лежащая в основе всех методов поддержания характеристик QoS заключается в следующем: *общая производительность каждого ресурса должна быть разделена между разными классами трафика неравномерно*.

Можно ввести более чем два класса обслуживания и стараться, чтобы каждый класс работал на своей части кривой задержек. Если такая задача решена, то можно обеспечить улучшение характеристик QoS за счет других методов, например снижая пульсацию трафика. Осталось выяснить, каким образом можно обеспечить такие условия для разных классов трафика в каждом узле сети.

Техника управления очередями

Техника управления очередями нужна для работы в периоды временных перегрузок, когда сетевое устройство не справляется с передачей пакетов на выходной интерфейс в том темпе, в котором они поступают. Если причиной перегрузки является недостаточная производительность процессорного блока сетевого устройства, то необработанные пакеты временно накапливаются во входной очереди соответствующего входного интерфейса. Очередей к входному интерфейсу может быть несколько, если мы дифференцируем запросы на обслуживание по нескольким классам. В том же случае, когда причина перегрузки заключается в ограниченной пропускной способности выходного интерфейса, пакеты временно сохраняются в выходной очереди (или очередях) этого интерфейса.

Очередь FIFO

В **очереди FIFO** в случае перегрузки все пакеты помещаются в *одну* общую очередь и выбираются из нее в том порядке, в котором поступили. Во всех устройствах с коммутацией пакетов алгоритм FIFO используется по умолчанию, так что такая очередь также обычно называется очередью «по умолчанию». Достоинствами этого подхода является простота реализации и отсутствие потребности в конфигурировании. Однако ему присущ и коренной недостаток — *невозможность дифференцированной обработки пакетов различных потоков*. Все пакеты стоят в общей очереди на равных основаниях. Вместе оказываются и пакеты чувствительного к задержкам голосового трафика, и пакеты нечувствительного к задержкам, но очень интенсивного трафика резервного копирования, длительные пульсации которого могут надолго задержать голосовой пакет.

Приоритетное обслуживание

Очереди с приоритетным обслуживанием очень популярны во многих областях вычислительной техники, в частности в операционных системах, когда одним приложениям нужно отдать предпочтение перед другими при обработке их в мультипрограммной смеси. Применяются эти очереди и для преимущественной по сравнению с другими обработки одного класса трафика.

Механизм приоритетного обслуживания основан на разделении всего сетевого трафика на небольшое количество классов и последующего назначения каждому классу некоторого числового признака — **приоритета**.

Классификация трафика представляет собой отдельную задачу. Пакеты могут разбиваться на приоритетные классы на основании различных признаков: адреса назначения, адреса источника, идентификатора приложения, генерирующего этот трафик, любых других комбинаций признаков, которые содержатся в заголовках пакетов. Правила классификации пакетов представляют собой часть политики администрирования сети.

Точка классификации трафика может размещаться в каждом коммуникационном устройстве. Более масштабируемое решение — размещение функций классификации трафика в одном или нескольких устройствах, расположенных на границе сети (например, в коммутаторах корпоративной сети, к которым подключаются компьютеры пользователей, или во входных маршрутизаторах сети поставщика услуг). В этом случае необходимо специальное поле в пакете, в котором можно запомнить назначенное значение приоритета,

чтобы им могли воспользоваться остальные сетевые устройства, обрабатывающие трафик после классифицирующего устройства. Такое поле имеется в заголовке многих протоколов. В тех же случаях, когда специального поля приоритета в заголовке нет, разрабатывается дополнительный протокол, который вводит новый заголовок с таким полем (так произошло, например, с протоколом Ethernet).

Приоритеты могут назначаться не только коммутатором или маршрутизатором, но и приложением на узле-отправителе. Необходимо также учитывать, что если в сети отсутствует централизованная политика назначения приоритетов, каждое сетевое устройство может не согласиться с приоритетом, назначенным данному пакету в другой точке сети. В этом случае оно перепишет значение приоритета в соответствии с локальной политикой, принятой непосредственно на данном устройстве.

В сетевом устройстве, поддерживающем приоритетное обслуживание, имеется *несколько* очередей (буферов) — по одной для каждого приоритетного класса. Пакет, поступивший в период перегрузок, помещается в очередь, соответствующую его приоритетному классу[1]. На рис. 7.9 приведен пример использования четырех приоритетных очередей с высоким, средним, нормальным и низким приоритетами. До тех пор пока из более приоритетной очереди не будут выбраны все имеющиеся в ней пакеты, устройство не переходит к обработке следующей менее приоритетной очереди. Поэтому пакеты с низким приоритетом обрабатываются только тогда, когда пустеют все вышестоящие очереди: с высоким, средним и нормальным приоритетами.

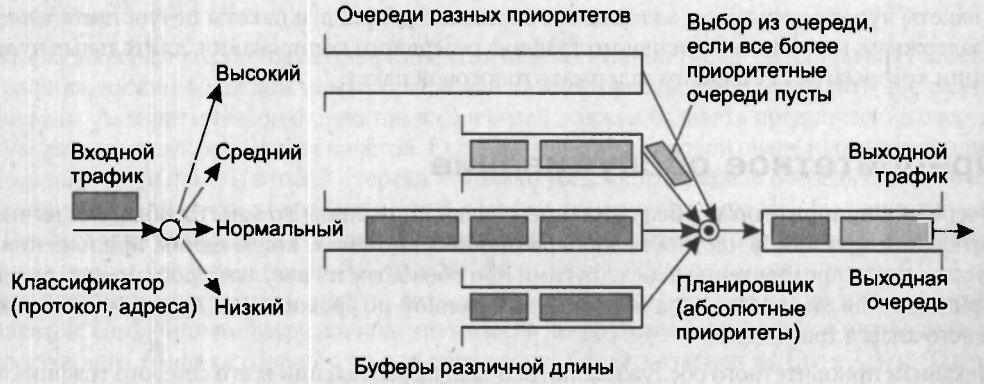


Рис. 7.9. Приоритетные очереди

Размер буфера сетевого устройства определяет максимальную длину очереди ожидающих обслуживания пакетов, если пакет поступает при заполненном буфере, то он просто отбрасывается. Обычно по умолчанию всем приоритетным очередям отводятся одинаковые буферы, но многие устройства разрешают администратору назначать каждой очереди буфер индивидуального размера. Размер буфера определяется в идеальном случае таким образом, чтобы его хватало с некоторым запасом для хранения очереди среднестатистической длины. Однако установить это значение достаточно сложно, так как оно изменяется

[1] Иногда несколько очередей изображают в виде одной очереди, в которой находятся заявки различных классов. Если заявки выбираются из очереди в соответствии с их приоритетами, то это просто другое представление одного и того же механизма.

в зависимости от нагрузки сети, поэтому требуется постоянное и длительное наблюдение за работой сети. В общем случае, чем выше значимость трафика для предприятия, чем больше его интенсивность и пульсации, тем больший размер буфера требуется этому трафику. В примере, приведенном на рис. 7.9, для трафика высшего и нормального приоритетов выбраны большие размеры буферов, а для остальных двух классов — меньшие. Мотивы принятого решения для высшего приоритета очевидны, а трафик нормального приоритета имеет, очевидно, высокую интенсивность и значительный коэффициент пульсаций.

Приоритетное обслуживание очередей обеспечивает высокое качество обслуживания для пакетов из самой приоритетной очереди. Если средняя интенсивность их поступления в устройство не превосходит пропускной способности выходного интерфейса (и производительности внутренних продвигающих блоков самого устройства), то пакеты высшего приоритета всегда получают ту пропускную способность, которая им нужна. Уровень задержек высокоприоритетных пакетов также минимален. Однако он не нулевой и зависит в основном от характеристик потока этих пакетов — чем выше пульсации потока и его интенсивность, тем вероятнее возникновение очереди, образованной пакетами данного высокоприоритетного потока. Трафик всех остальных приоритетных классов почти прозрачен для пакетов высшего приоритета. Слово «почти» относится к ситуации, когда высокоприоритетный пакет вынужден ждать завершения обслуживания низкоприоритетного пакета, если его приход совпадает по времени с началом продвижения низкоприоритетного пакета на выходной интерфейс. Этот эффект иллюстрирует рис. 7.10, на котором показано, что после разделения всего трафика на приоритетный и обычный (то есть здесь имеются две очереди), коэффициент использования для приоритетного трафика снизился с 50 до 15 %.

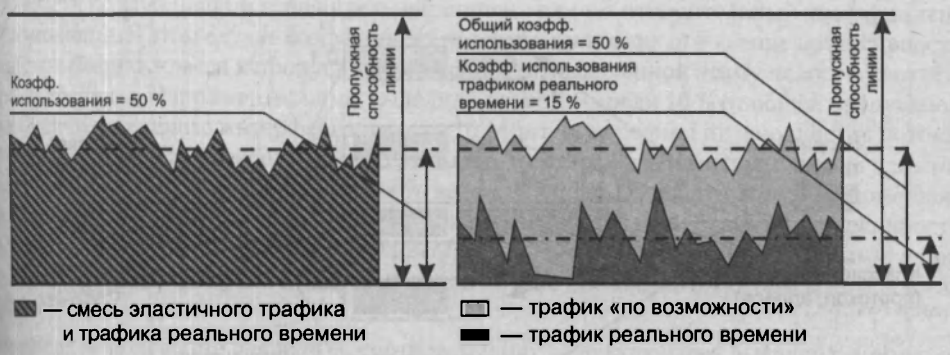


Рис. 7.10. Снижении коэффициента использования линии для приоритетного трафика: *а* — весь трафик обслуживается одной очередью; *б* — трафик реального времени обслуживается приоритетной очередью, а остальной трафик — очередью по умолчанию

Что же касается остальных приоритетных классов, то качество их обслуживания будет ниже, чем у пакетов самого высокого приоритета, причем уровень снижения может быть очень существенным. Если коэффициент нагрузки выходного интерфейса, определяемый только трафиком высшего приоритетного класса, приближается в какой-то период времени к единице, то трафик остальных классов просто на это время замораживается. Поэтому

приоритетное обслуживание 0обычно применяется для чувствительного к задержкам класса трафика, имеющего небольшую интенсивность. При таких условиях обслуживание этого класса не слишком ущемляет обслуживание остального трафика. Например, голосовой трафик чувствителен к задержкам, но его интенсивность обычно не превышает 8–16 Кбит/с, так что при назначении ему высшего приоритета ущерб остальным классам трафика оказывается не очень значительным.

Взвешенные очереди

Механизм взвешенных очередей разработан для того, чтобы можно было предоставить всем классам трафика определенный минимум пропускной способности. Под *весом* данного класса понимается процент предоставляемой классу трафика пропускной способности от полной пропускной способности выходного интерфейса.

При взвешенном обслуживании, так же, как и при приоритетном, трафик делится на несколько классов, и для каждого класса ведется отдельная очередь пакетов. Но с каждой очередью связывается *не приоритет, а процент пропускной способности* ресурса, гарантируемый данному классу трафика при перегрузках этого ресурса. Для входного потока таким ресурсом является процессор, а для выходного (после выполнения коммутации) – выходной интерфейс.

ПРИМЕР

Показанное на рис. 7.11 устройство для пяти классов трафика поддерживает пять очередей к выходному интерфейсу коммутатора. Этим очередям при перегрузках выделяется соответственно 10, 10, 30, 20 и 30 % пропускной способности выходного интерфейса.

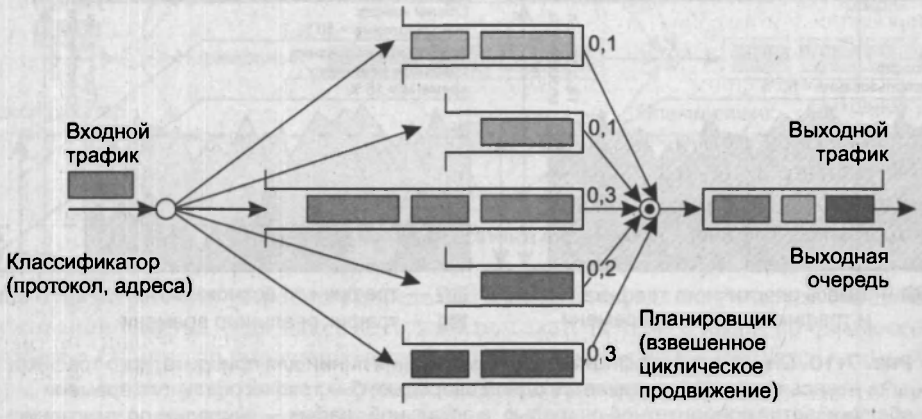


Рис. 7.11. Взвешенные очереди

Достигается поставленная цель за счет того, что очереди обслуживаются последовательно и циклически, и в каждом цикле обслуживания из каждой очереди выбирается такое число байтов, которое соответствует весу данной очереди. Так, если цикл просмотра очередей в рассматриваемом примере равен одной секунде, а скорость выходного интерфейса составляет 100 Мбит/с, то при перегрузках в каждом цикле первой очереди уделяется 10 % времени,

то есть 100 мс и выбирается 10 Мбит данных, из второй — тоже 10 Мбит, из третьей — 30 Мбит, из четвертой — 20 Мбит, из пятой — 30 Мбит.

В результате каждому классу трафика достается гарантированный минимум пропускной способности, что во многих случаях является более желательным результатом, чем подавление низкоприоритетных классов высокоприоритетным.

Так как данные выбираются из очереди пакетами, а не битами, то реальное распределение пропускной способности между классами трафика всегда немного отличается от планируемого. Так, в предыдущем примере вместо 10 % первый класс трафика мог бы получать при перегрузках 9 или 12 %. Чем больше время цикла, тем точнее соблюдаются требуемые пропорции между классами трафика, так как из каждой очереди выбирается большее число пакетов, и влияние размера каждого пакета усредняется.

В то же время длительный цикл приводит к большим задержкам передачи пакетов. Так, при выбранном нами для примера цикле в одну секунду задержка может составить одну секунду и больше — ведь арбитр возвращается к каждой очереди не чаще, чем раз в секунду, кроме того, в очереди может находиться более одного пакета. Поэтому при выборе времени цикла нужно обеспечить баланс между точностью соблюдения пропорций пропускной способности и стремлением к снижению задержки.

Для нашего примера время цикла в 1000 мкс является примером такого баланса. С одной стороны, оно обеспечивает обслуживание очереди каждого класса каждые 1000 мкс, а значит – более низкий уровень задержек. С другой стороны, этого времени достаточно, чтобы выбрать из каждой очереди в среднем по несколько пакетов (первой очереди в нашем примере будет отводиться 100 мкс, что достаточно для передачи в выходной канал одного пакета Fast Ethernet или десяти пакетов Gigabit Ethernet).

На уровень задержек и вариации задержек пакетов для некоторого класса трафика при взвешенном обслуживании в значительной степени влияет **относительный коэффициент использования**. В этом случае коэффициент подсчитывается как отношение интенсивности входного трафика класса к пропускной способности, выделенной этому классу в соответствии с его весом. Например, если мы выделили первой очереди 10 % от общей пропускной способности выходного интерфейса, то есть 10 Мбит/с, а средняя интенсивность потока, который попадает в эту очередь, равна 3 Мбит/с/, то коэффициент использования для этого потока составит 3/10 = 0,3. Зависимость на рис. 7.5 показывает, что задержки при таком значении коэффициента использования будут незначительными. Если бы интенсивность входного потока этой очереди была 9 Мбит/с, то очереди были бы значительными, а при превышении предела 10 Мбит/с часть пакетов потока постоянно бы отбрасывалась из-за переполнения очереди.

Качественное поведение очереди и, соответственно, задержек здесь выглядит примерно так же, как и в случае очереди FIFO — чем меньше коэффициент загрузки, тем меньше средняя длина очереди и тем меньше задержки.

Как и для приоритетного обслуживания, при взвешенном обслуживании администратор может назначать разным классам очередей буферы разных размеров. Уменьшение размеров буферов для очередей ведет к росту числа потерь пакетов при перегрузках, но зато снижает время ожидания для тех пакетов, которые не были отброшены и попали в очередь.

Еще одним вариантом взвешенного обслуживания является **взвешенное справедливое обслуживание** (Weighted Fair Queuing, WFQ). В случае подобного обслуживания пропускная способность ресурса делится между всеми потоками поровну, то есть «справедливо».

Взвешенное обслуживание обеспечивает требуемые соотношения между интенсивностями трафика различных очередей только *в периоды перегрузок*, когда каждая очередь постоянно заполнена. Если же какая-нибудь из очередей пуста (то есть для трафика данного класса текущий период не является периодом перегрузки), то при просмотре очередей она игнорируется, и ее время обслуживания распределяется между остальными очередями в соответствии с их весом. Поэтому в отдельные периоды трафик определенного класса может обладать большей интенсивностью, чем соответствующий процент от пропускной способности выходного интерфейса.

Комбинированные алгоритмы обслуживания очередей

Каждый из описанных подходов имеет свои достоинства и недостатки. Приоритетное обслуживание, обеспечивая минимальный уровень задержек для очереди наивысшего приоритета, не дает никаких гарантий в отношении средней пропускной способности для трафика очередей более низких приоритетов. Взвешенное обслуживание обеспечивает заданное распределение средней пропускной способности, но не учитывает требований к задержкам.

Существуют **комбинированные алгоритмы обслуживания очередей**. В наиболее популярном алгоритме подобного рода поддерживается одна приоритетная очередь, а остальные очереди обслуживаются в соответствии со взвешенным алгоритмом. Обычно приоритетная очередь используется для чувствительного к задержкам трафика, а остальные — для эластичного трафика нескольких классов. Каждый класс эластичного трафика получает некоторый минимум пропускной способности при перегрузках. Этот минимум вычисляется как процент от пропускной способности, оставшейся от приоритетного трафика. Очевидно, что нужно как-то ограничить приоритетный трафик, чтобы он не поглощал всю пропускную способность ресурса. Обычно для этого применяются механизмы кондиционирования трафика, которые рассматриваются далее.

Механизмы кондиционирования трафика

Механизмы кондиционирования трафика контролируют текущие параметры потоков трафика, такие как его средняя скорость и пульсация. Как мы помним, основной идеей методов QoS является выделение определенной доли пропускной способности определенным потокам трафика, при этом величина этой доли должна быть достаточной для того, чтобы коэффициент использования ресурса для потока был достаточно низким, и соответственно качество обслуживания потока было удовлетворительным. Очереди с различными алгоритмами обслуживания позволяют реализовать только одну часть этой идеи — они выделяют определенную долю пропускной способности некоторому потоку пакетов. Однако остается вторая часть задачи — удержание скорости потока в определенных пределах с целью обеспечить желаемый коэффициент использования пропускной способности, которая выделена потоку с помощью некоторой очереди. Если же скорость потока не будет соответствовать ожидаемой, то вся работа по выделению потоку пропускной способности не приведет к желаемому результату, так как коэффициент использования этой пропускной способности будет отличаться от ожидаемого, и нужное качество обслуживания достигнуто не будет.

Механизмы кондиционирования трафика являются своего рода контрольно-пропускными пунктами, которые проверяют трафик на входе в коммутатор (или формируют трафик на выходе из него — для чего это нужно, мы рассмотрим немного далее). Существует несколько механизмов кондиционирования трафика.

Классификация трафика

Классификация трафика представляет собой элемент QoS, позволяющий определить, какие пакеты нужно отправить в ту или иную очередь. Классификация обычно выполняется средствами фильтрации трафика, имеющимися в коммутаторах и маршрутизаторах пакетных сетей; для этих средств используется также такое название, как **списки контроля доступа** (Access Control List, ACL)[1]. Для классификации используются различные признаки пакетов, например адреса назначения и источника, тип протокола транспортного или прикладного уровня. Мы уже упоминали классификацию трафика при описании приоритетных и взвешенных очередей, так как без этого механизма кондиционирования трафика невозможно задействовать различные очереди к одному и тому же выходному интерфейсу.

Профилирование

Профилирование представляет собой меру принудительного воздействия на трафик, которая служит для ограничения скорости потока пакетов. Профилирование обеспечивает соответствие потока пакетов заданному скоростному профилю; в качестве параметров профиля обычно выбирается средняя скорость потока пакета, измеренная на определенном интервале времени[2]. Пакеты, которые не укладываются в заданный профиль, либо отбрасываются, либо деквалифицируются, то есть помещаются в класс обслуживания с более низкими привилегиями, например переводятся из приоритетного класса в стандартный класс, обслуживаемый «по возможности». Англоязычное название операции профилирования — policing[3] — кажется более жестким и, возможно, дает более адекватное представление о характере действий.

Профилирование обычно применяют для ограничения трафика, поступающего в приоритетную очередь, так как этот механизм является единственно возможным средством предотвращения ситуации вытеснения всего остального трафика приоритетным трафиком.

Рисунок 7.12 иллюстрирует действие механизма профилирования, показывая значения скорости трафика, измеренные на достаточно малых интервалах времени до и после профилирования. Как видно из рисунка, отбрасывание пакетов при профилировании приводит к удержанию скорости потока на заданном уровне в те интервалы времени, когда скорость входящего потока превосходит этот предел, и к сохранению исходной скорости в остальные периоды.

---

[1] Их не нужно путать со средствами контроля допуска (admission control) трафика, которые также используются в системах обеспечения качества обслуживания, но имеют другое назначение (см. далее).

[2] Применяются и более сложные варианты профилирования, например, учитывающие среднюю и пиковые скорости.

[3] Дословно — поддерживать порядок полицейскими средствами.

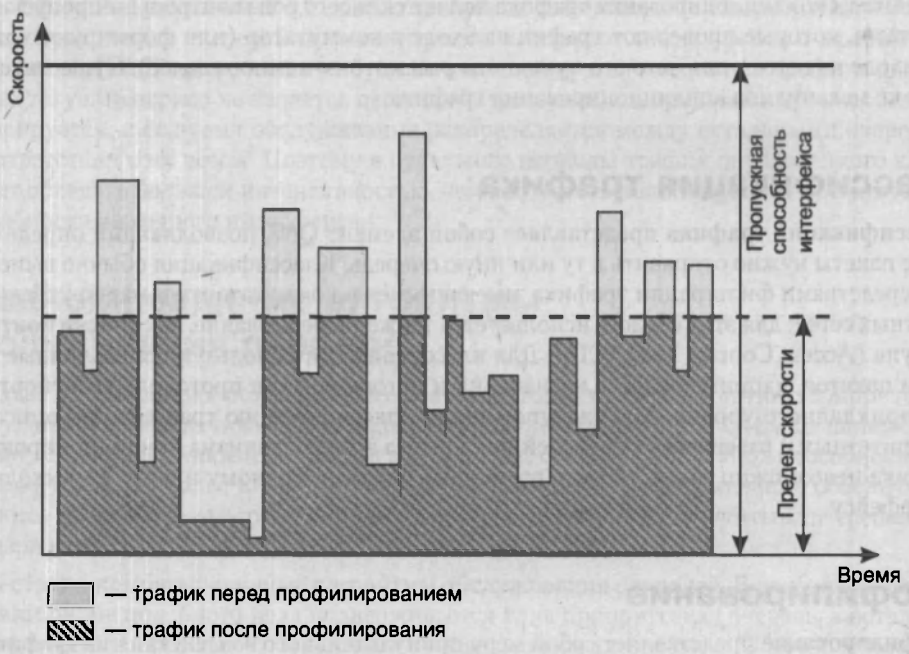


Рис. 7.12. Эффект профилирования — отбрасывание избыточного трафика

Формирование трафика

Формирование трафика — это процесс намеренной задержки некоторых пакетов из общего потока в целях удержания средней скорости трафика в некоторых заданных пределах. Формирование трафика напоминает профилирование, так как имеет схожую цель — ограничение скорости трафика, но достигается эта цель другим способом. Вместо того чтобы отбрасывать избыточные пакеты (то есть те, передача которых могла бы привести к превышению лимита скорости), механизм формирования трафика задерживает пакеты-нарушители так, что результирующая скорость оказывается в заданных пределах. Эффект формирования трафика[1] иллюстрирует рис. 7.13. Из рисунка видно, что скорость трафика сглаживается, так как избыточные пакеты не отбрасываются, а передаются с задержкой в другие интервалы времени. Тем самым скорость исходного потока снижается в течение периодов времени с избыточным трафиком и растет в тех последующих интервалах, в которых она оказывается меньше установленного предела.

Обычно путем формирования обрабатывают трафик, исходящий из коммутатора или маршрутизатора. Это делается в тех случаях, когда известно, что некоторое коммуникационное устройство далее по маршруту следования потока пакетов применяет профилирование. Профиль формирования трафика выбирается равным профилю профилируемого трафика, это гарантирует отсутствие потерь трафика из-за отбрасывания избыточных пакетов.

---

[1] На заднем плане рисунка показана скорость результирующего пакета, а на переднем — скорость исходного потока (полупрозрачным заполнением).

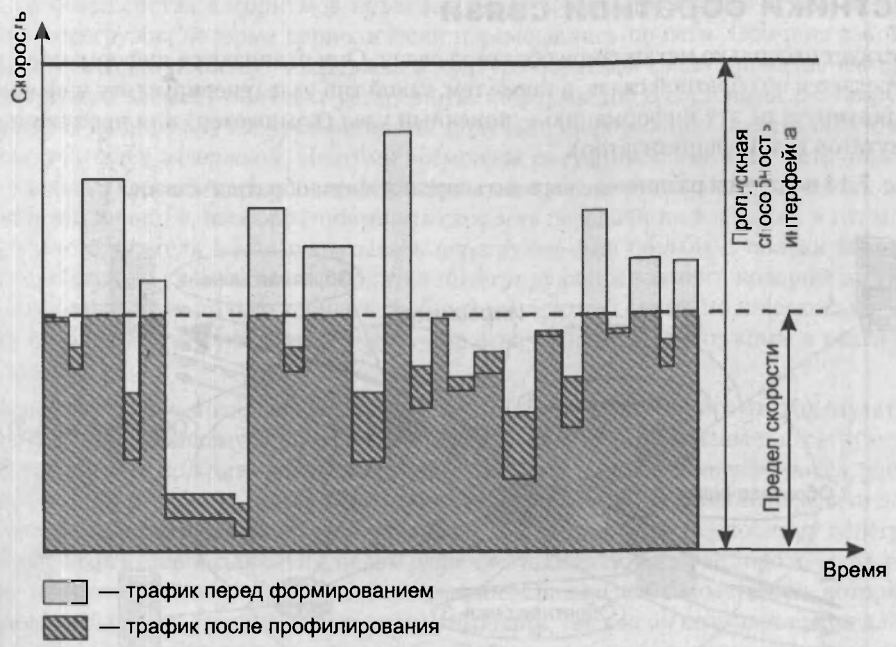


Рис. 7.13. Эффект формирования трафика — сглаживание

Механизмы кондиционирования трафика могут поддерживаться каждым узлом сети или реализовываться только в пограничных устройствах. Последний вариант часто используют поставщики услуг, кондиционируя трафик своих клиентов.

Обратная связь

Назначение

Алгоритмы управления очередями и кондиционирования трафика не предотвращают перегрузок, а лишь некоторым «справедливым» образом в условиях дефицита перераспределяют ресурсы между различными потоками или классами трафика. Алгоритмы управления очередями относятся к механизмам **управления перегрузкой** (congestion management), которые начинают работать, когда сеть уже перегружена.

Существует другой класс средств, которые носят название механизмов **предотвращения перегрузки** (congestion avoidance). Этот механизм основан на использовании *обратной связи*, с помощью которого перегруженный узел сети, реагируя на перегрузку, просит предыдущие узлы, расположенные вдоль маршрута следования потока (или потоков, принадлежащих к одному классу), временно снизить скорость трафика. После того как перегрузка в данном узле исчезнет, он посылает другое сообщение, разрешающее повысить скорость передачи данных.

Таким образом, при возникновении перегрузки механизм предотвращения перегрузок за счет обратной связи *временно* снижает нагрузку. Существует и другое название этого механизма – **активное управление очередями**.

Участники обратной связи

Существует несколько механизмов обратной связи. Они отличаются информацией, которая передается по обратной связи, а также тем, какой тип узла генерирует эту информацию и кто реагирует на эту информацию — конечный узел (компьютер) или промежуточный (коммутатор или маршрутизатор).

На рис. 7.14 показаны различные варианты организации обратной связи.

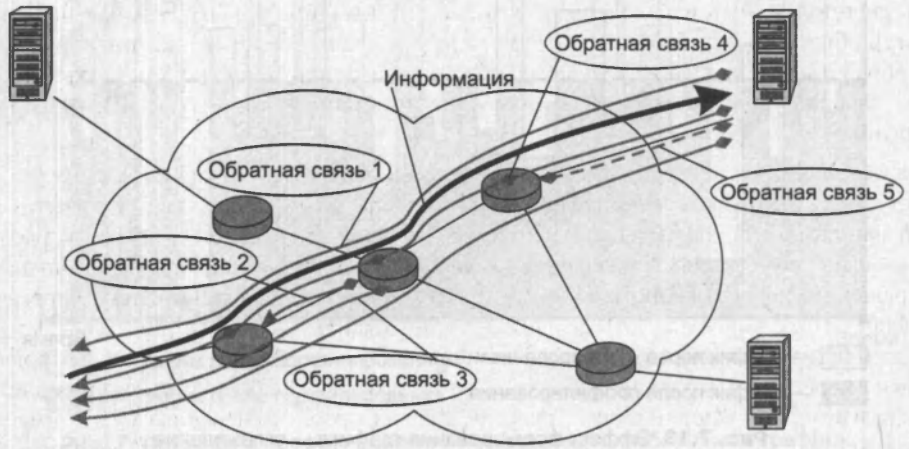


Рис. 7.14. Участники обратной связи

Обратная связь 1 организована между двумя конечными узлами сети. Этот вариант обеспечивает наиболее радикальное снижение нагрузки на сеть, так как только конечный узел может снизить скорость поступления информации в сеть. Однако этот вид обратной связи не относят к методам управления перегрузкой, так как его назначение — борьба с перегрузками узла назначения, а не с перегрузками сетевых устройств. Принципиально эта та же самая проблема, так как она является следствием временного превышения скорости поступления пакетов в ресурс над скоростью обработки этих пакетов. Только ресурсом в данном случае выступает не коммутатор сети, а конечный узел. Но традиционно за этим видом обратной связи закрепилось собственное название — **контроль потока**. Устройства сети не принимают участие в работе этого вида механизма обратной связи, они только передают соответствующие сообщения между конечными узлами. Несмотря на разные названия, в методах управления перегрузкой и контроля потока используются общие механизмы.

При организации обратной связи важно учитывать время передачи информации по сети. В высокоскоростных глобальных сетях за время передачи сообщения о перегрузке узла назначения узел-источник может успеть передать через сеть тысячи пакетов, так что перегрузка не будет ликвидирована вовремя. Из теории автоматического управления известно, что задержки в контуре обратной связи могут приводить ко многим нежелательным эффектам, прямо противоположным первоначальным целям. Например, в системе могут начаться колебательные процессы, и она никогда не сможет прийти в равновесное состояние. Подобные явления наблюдались на ранней стадии развития Интернета, когда

из-за несовершенства алгоритмов обратной связи и маршрутизации в нем возникали участки перегрузок, которые периодически перемещались по сети. Причина такой проблемы интуитивно понятна — задержка в контуре обратной связи приводит к тому, что регулирующий элемент получает устаревшую информацию о состоянии регулируемого элемента. В данном случае узел-источник получает информацию о состоянии очереди узла-получателя с задержкой. Поэтому возможны ситуации, когда узел-источник начинает снижать скорость передачи информации, хотя в действительности очереди в узле-получателе уже нет, и, наоборот, повышать скорость передачи информации в тот момент, когда узел-получатель начал испытывать перегрузку. Для борьбы с такими явлениями в контур обратной связи обычно вводится интегрирующий элемент, который на каждом шаге обрабатывает не только текущее сообщение обратной связи, но и несколько предыдущих сообщений, что позволяет учесть динамику изменения ситуации и реагировать адекватно.

Обратная связь 2 организована между двумя соседними коммутаторами. Коммутатор сообщает соседу, находящемуся выше по течению потока, что он испытывает перегрузку и его буфер заполнился до критической величины. Получив такое сообщение, сосед, расположенный выше по течению, должен снизить на некоторое время скорость передачи данных в направлении перегруженного коммутатора и тем самым решить проблему перегрузки. Это менее эффективное для сети в целом решение, так как поток будет продолжать течь от узла-источника с той же скоростью, что и раньше. Однако для коммутатора, который испытывает перегрузку, это является хорошим выходом, так как он получает время для того, чтобы разгрузить переполнившуюся очередь. Правда, проблема переносится в коммутатор, расположенный выше по течению, в котором теперь может возникнуть перегрузка, так как он начинает передавать данные из своего буфера с меньшей скоростью. Достоинством описанного метода является низкая задержка обратной связи, так как узлы являются соседями (если они, конечно, не соединены спутниковым каналом).

Обратная связь 3 организована между промежуточным коммутатором и узлом-источником. Сообщения обратной связи хотя и передаются несколькими коммутаторами сети в направлении узла-источника, но они на него не реагируют.

В *обратной связи 4*, как и в обратной связи 1, сообщение о перегрузке порождается узлом-получателем и передается узлу-источнику. Однако имеется и важное отличие: в данном случае каждый промежуточный коммутатор реагирует на это сообщение. Во-первых, он снижает скорость передачи данных в направлении узла назначения, во-вторых, он может изменить содержание сообщения. Например, если узел назначения просит снизить скорость до 30 Мбит/с, то промежуточный коммутатор может снизить эту величину до 20 Мбит/с, оценив состояние своего буфера. Кроме того, породить сообщение обратной связи может любой коммутатор сети, а не только узел назначения.

При описании различных вариантов организации обратной связи мы подразумевали, что сообщение о перегрузке идет в направлении, обратном направлению передачи пользовательской информации (собственно, поэтому этот механизм так и называется). Однако некоторые коммуникационные протоколы не предусматривают возможности генерации подобных сообщений промежуточными узлами. В таких условиях часто применяют искусственный прием — передача сообщения о перегрузке узлу назначения, который преобразует его в сообщение обратной связи и отправляет в нужном направлении, то есть в направлении источника. Этот вариант показан на рисунке как *обратная связь 5*.

Информация обратной связи

В применяемых сегодня методах обратной связи используются следующие основные типы сообщений:

- признак перегрузки;
- максимальная скорость передачи;
- максимальный объем данных (кредит);
- косвенные признаки.

Признак перегрузки не говорит о степени перегруженности сети или узла, он только фиксирует факт наличия перегрузки. Реакция узла, получившего такое сообщение, может быть разной. В некоторых протоколах узел обязан прекратить передачу информации в определенном направлении до тех пор, пока не будет получено другое сообщение обратной связи, разрешающее продолжение передачи. В других протоколах узел ведет себя адаптивно, он снижает скорость на некоторую величину и ожидает реакции сети. Если сообщения с признаком перегрузки продолжают поступать, то он продолжает снижение скорости.

Во втором типе сообщений указывается **максимальная скорость передачи**, то есть порог скорости, который должен соблюдать источник или промежуточный узел, расположенный выше по течению потока. В этом случае обязательно нужно учитывать время передачи сообщения по сети, чтобы исключить колебательные процессы в сети и обеспечить нужную скорость реакции на перегрузку. Поэтому в территориальных сетях такой способ обычно реализуется силами всех коммутаторов сети (обратная связь 4 в нашем примере).

Сообщение о **максимальном объеме данных** используется в широко применяемом в пакетных сетях алгоритме скользящего окна (подробнее о нем рассказывается в главе 17). Этот алгоритм позволяет не только обеспечивать надежную передачу данных, но и реализовать обратную связь для контроля потока между конечными узлами. Параметром, несущим информацию обратной связи, является «окно» — число, тесно связанное с текущим размером свободного пространства в буфере принимающего узла. Окно также называют **кредитом**, который принимающий дает передающему узлу. Передающий узел может с любой скоростью передать объем информации (или определенное количество пакетов, если окно измеряется в пакетах), соответствующий кредиту. Но если кредит исчерпан, то передающий узел не имеет права передавать информацию, пока не получит следующий кредит. При перегрузках принимающий узел уменьшает размер окна, тем самым снижая нагрузку. Если эффект перегрузки исчезает, то принимающий узел увеличивает размер окна. Недостатком этого алгоритма является то, что он работает только в протоколах с установлением соединения.

И, наконец, в некоторых случаях передающий узел определяет, что принимающий узел (или узлы) испытывает перегрузку, по некоторым **косвенным признакам**, без получения сообщения обратной связи. Такими косвенными признаками могут быть факты потери пакетов. Для того чтобы протокол мог обнаруживать факты потерь пакетов, это должен быть протокол с установлением соединения. Тогда истечение тайм-аута или приход дубликата положительной квитанции косвенно свидетельствует о том, что пакет потерян. Однако потеря пакета не всегда свидетельствует о перегрузке сети. Перегрузка сети — это только одна из возможных причин потери пакета, другой причиной может быть ненадежная работа коммуникационных устройств (отказы оборудования, искажения данных из-за помех). Тем не менее так как реакция на перегрузки и ненадежную работу сети должна быть одинаковой

и состоять в снижении скорости передачи, то неоднозначность причины потери пакета не является проблемой.

Примером протокола, использующего неявную информацию о перегрузках, является протокол TCP. Этот протокол с помощью явной информации обратной связи (о размере окна) осуществляет контроль потока, а с помощью неявной (потери пакетов, дубликаты квитанций) — управляет перегрузкой.

Резервирование ресурсов

Резервирование ресурсов и контроль допуска

Рассмотренные методы поддержания качества обслуживания ориентированы в основном на борьбу с перегрузками или предотвращение их в пределах отдельного узла сети. Вместе с тем понятно, что для обеспечения гарантированного уровня качества обслуживания некоторого потока пакетов необходимо скоординированное применение этих методов на всем пути следования потока через сеть. Такой координирующей процедурой является процедура резервирования ресурсов сети для определенного потока. Эта процедура позволяет настроить все механизмы поддержания качества обслуживания вдоль следования потока таким образом, чтобы поток с некоторыми заданными характеристиками скорости был обслужен с заданными характеристиками QoS.

Основная идея процедуры состоит в том, что каждому узлу сети вдоль маршрута следования потока задается вопрос, может ли этот узел обслужить некоторый новый поток с заданными характеристиками QoS, если известны предельные характеристики скорости потока, такие как средняя и пиковая скорости? Каждый узел при ответе на этот вопрос должен оценить свои возможности, то есть проверить, достаточно ли у него свободных ресурсов, чтобы принять на обслуживание новый поток и обслуживать его качественно. При положительном ответе узел должен некоторым образом зарезервировать часть своих ресурсов для данного потока, чтобы при поступлении пакетов потока на входные интерфейсы использовать эти ресурсы для их обслуживания с гарантированным уровнем качества.

В общем случае каждый узел самостоятельно решает, какие ресурсы он должен резервировать для обслуживания некоторого потока с заданным качеством. Как показывает практика, основным ресурсом, требуемым для качественного обслуживания пакетов, является пропускная способность интерфейса, через который пакеты потока покидают узел. Поэтому в дальнейшем мы будем, несколько упрощая действительное положение дел, употреблять формулировку «резервирование пропускной способности» вместо «резервирование ресурсов».

Смысл резервирования состоит в том, чтобы ограничить уровень перегрузок определенного потока или нескольких потоков некоторой приемлемой величиной. Эта величина должна быть такой, чтобы механизмы QoS (управления очередями, кондиционирования трафика и обратной связи), применяемые в узлах сети, справлялись с кратковременными небольшими перегрузками и обеспечивали требуемые значения характеристик QoS.

Однако что же означает резервирование пропускной способности в сетях с коммутацией пакетов? Ранее мы не встречались с таким механизмом, все предыдущие объяснения работы сетевых устройств обходились без него. Дело в том, что этот механизм не является

обязательным, а используется только в тех случаях, когда требуется гарантированное выполнение требований качества обслуживания пакетов.

Резервирование пропускной способности в сетях с коммутацией пакетов похоже на аналогичную процедуру в сетях с коммутацией каналов тем, что определенному потоку данных назначается определенная часть пропускной способности линии связи. Однако это назначение здесь гораздо более гибкое — если отведенная пропускная способность в какой-то период времени недоиспользуется потоком, то она может быть передана другим потокам. Это обстоятельство позволяет более эффективно расходовать пропускную способность линий связи, но приводит к эффекту постепенной деградации качества транспортного сервиса из-за перегрузок и очередей вместо простого отказа в обслуживании, который имеет место в сети с коммутацией каналов, когда пропускной способности оказывается недостаточно для обслуживания некоторого потока. Цель гибкого резервирования — обеспечить поток зарезервированной пропускной способностью в те периоды, когда она ему нужна вся, то есть в периоды перегрузок. Другим отличием резервирования в пакетных сетях является то обстоятельство, что оно может выполняться не только «из конца в конец», но и для каких-то отдельных узлов по маршруту потока, однако этот случай не может гарантировать необходимый уровень характеристик QoS, так как перегрузка даже в одном узле может привести к задержкам и потерям пакетов.

Резервирование пропускной способности в пакетной сети «из конца в конец» начинается с операции, называемой **контролем допуска в сеть** потока, который просит зарезервировать для своего обслуживания некоторую пропускную способность сети между ее двумя конечными узлами. Эта операция состоит в проверке наличия доступной (то есть незарезервированной для других потоков) пропускной способности в каждом из узлов сети на протяжении всего маршрута следования потока (здесь мы не останавливаемся на проблеме поиска маршрута потока, она подробно рассматривается далее в разделе «Инжиниринг трафика»). Очевидно, что максимальная средняя скорость потока должна быть меньше, чем запрашиваемая пропускная способность, иначе поток будет обслужен с очень плохим качеством даже несмотря на то, что ему была зарезервирована некоторая пропускная способность.

Если результат контроля допуска положителен в каждом узле (случай, показанный на рис. 7.15), то сетевые устройства запоминают факт резервирования, чтобы при появлении пакетов данного потока распознать их и выделить им зарезервированную пропускную способность. Кроме того, при успешном резервировании доступная для резервирования (в будущем) пропускная способность уменьшается на величину, зарезервированную за данным потоком. Как видно из описания процедуры, для ее реализации необходимо знать маршрут следования потока, для которого выполняется резервирование. В сетях с распределенным принципом построения таблиц маршрутизации, когда каждое сетевое устройство самостоятельно определяет следующий по маршруту узел, выяснение маршрута может представлять достаточно сложную задачу, но мы оставим исследование этой проблемы до рассмотрения конкретных технологий в последующих главах, а пока будем считать, что маршрут каким-то образом нам известен.

Нужно подчеркнуть, что резервирование — это процедура, которая выполняется *перед тем*, как реальный трафик будет направлен в сеть.

Давайте теперь посмотрим, каким же образом выполняется собственно выделение пропускной способности потоку в моменты времени, когда его пакеты поступают на вход коммуникационного устройства $S2$, которое запомнило факт резервирования пропускной способности для потока $F1$ на выходном интерфейсе $P2$ (рис 7.16).

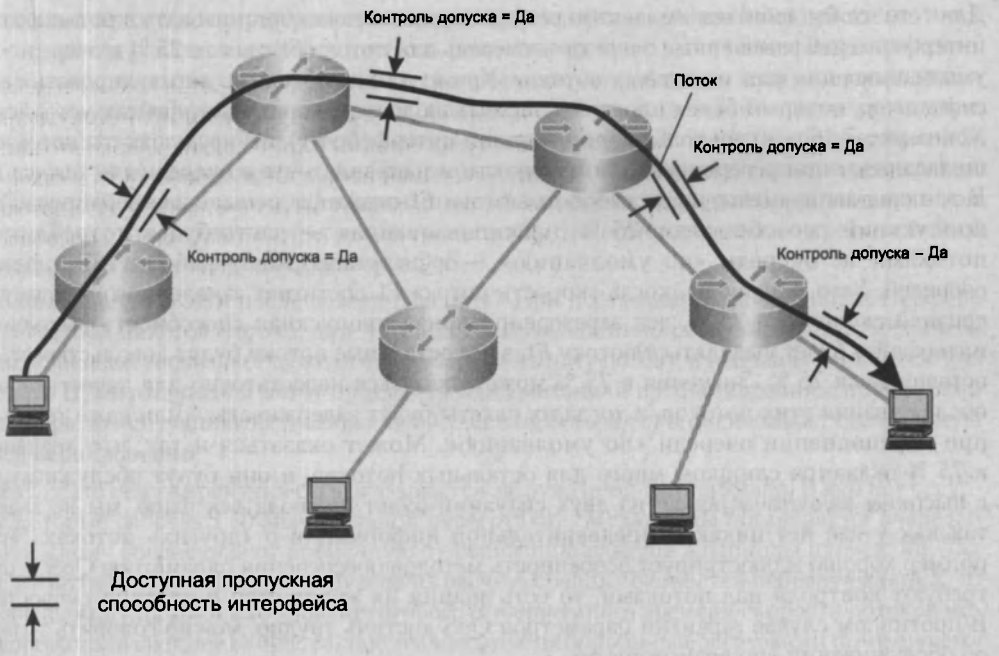


Рис. 7.15. Контроль допуска потока

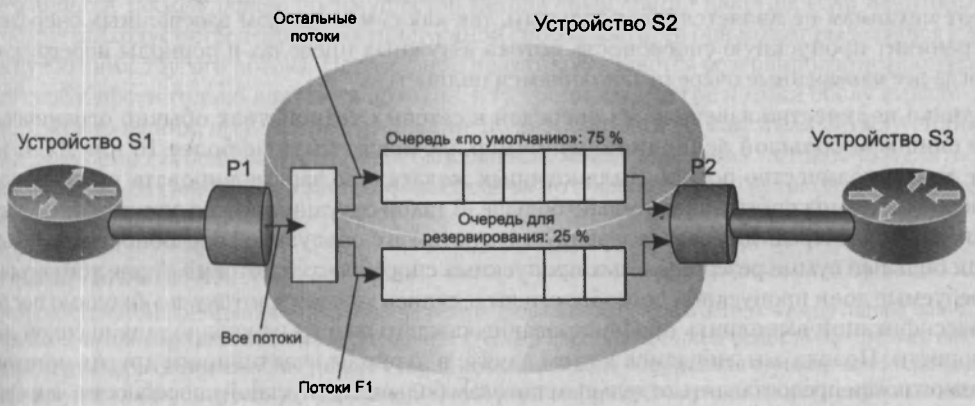


Рис. 7.16. Выделение зарезервированной пропускной способности

Такое выделение можно обеспечить разными способами, в нашем примере это будет сделано с использованием взвешенных очередей.

Пусть потоку *F1* при резервировании было выделено 25 % пропускной способности интерфейса *P2* (обычно резервирование можно выполнять как в абсолютных величинах, например в мегабитах в секунду, так и в процентах; это, собственно, детали реализации механизмов QoS в конкретных устройствах). Также для простоты будем считать, что резервирование было выполнено только для потока *F1*, в то же время для всех других потоков, которые проходят через выходной интерфейс *P2*, резервирования не производилось.

Для того чтобы добиться желаемого результата, достаточно организовать для выходного интерфейса две взвешенные очереди — очередь для потока *F*1 с весом 25 % и очередь «по умолчанию» для всех остальных потоков. Кроме того, необходимо активизировать *классификатор*, который будет проверять пакеты на всех входных интерфейсах устройства *S*2 (на рис. 7.16 показан только один входной интерфейс *P*1), отбирать пакеты потока *F*1 по заданным при резервировании признакам и направлять их в очередь для потока *F*1. В те периоды времени, когда скорость потока *F*1 окажется меньше зарезервированной пропускной способности в 25 %, неиспользованная ее часть будет потребляться потоками из очереди «по умолчанию» — в силу алгоритма работы взвешенных очередей. Зато в периоды, когда скорость потока *F*1 достигнет заявленного максимума средней скорости в 25 %, вся зарезервированная пропускная способность выходного интерфейса будет выделяться потоку *F*1, а все остальные потоки будут довольствоваться оставшимися 75 %. Значения в 75 % может оказаться недостаточно для качественного обслуживания этих потоков, и тогда их пакеты будут задерживаться или даже теряться при переполнении очереди «по умолчанию». Может оказаться и так, что значения в 75 % окажется слишком много для остальных потоков, и они будут обслуживаться с высоким качеством; какая из двух ситуаций будет наблюдаться чаще, мы не знаем, так как у нас нет никакой предварительной информации о «других» потоках. Этот пример хорошо иллюстрирует особенность методов обеспечения параметров QoS — они требуют контроля над потоками, то есть знания их маршрутов и средних скоростей. В противном случае гарантий параметров QoS достичь трудно, можно говорить только об обслуживании «по возможности».

В описанном примере не использован механизм профилирования трафика. При наличии отдельной взвешенной очереди для потока, зарезервировавшего пропускную способность, этот механизм не является обязательным, так как сам механизм взвешенных очередей ограничит пропускную способность потока в нужных пределах в периоды перегрузок, когда все взвешенные очереди заполняются полностью.

Однако количество взвешенных очередей в сетевых устройствах обычно ограничено не слишком большой величиной, например, их может быть не более 16 или 32. В то же время количество потоков, для которых желательно зарезервировать пропускную способность, может быть значительно больше. В такой ситуации можно организовать одну взвешенную очередь для всех резервируемых потоков с пропускной способностью, равной или большей сумме резервируемых пропускных способностей потоков. А для того чтобы требуемые доли пропускной способности выделялись каждому потоку, необходимо после классификации выполнить профилирование каждого потока на уровне запрошенной им скорости. Правда, мы лишаемся в этом случае в периоды неактивности других потоков возможности предоставлять отдельным потокам больше пропускной способности, чем они запросили, но это плата за масштабируемое решение, основанное на одной взвешенной резервируемой очереди.

Использование взвешенных очередей — не единственный вариант резервирования пропускной способности в пакетных сетях. Для той же цели можно задействовать приоритетные очереди. Применение приоритетной очереди может быть не только возможным, но и необходимым, если потоку помимо определенного уровня пропускной способности требуется обеспечить минимально возможный уровень задержек пакетов.

При использовании приоритетной очереди профилирование необходимо всегда, так как приоритетный механизм не обеспечивает ограничения скорости потока, как это делает механизм взвешенного обслуживания.

Нужно подчеркнуть, что резервирование приводит к ожидаемым результатам только в тех случаях, когда реальная скорость потоков, для которых было выполнено резервирование, оказывается не выше, чем пропускная способность, запрошенная при резервировании и реализованная при конфигурировании сетевых устройств. В противном случае результаты могут оказаться даже хуже, чем при наличии единственной очереди «по умолчанию» и обслуживании «по возможности». Так, если скорость потока окажется выше, чем предел, учитываемый механизмом профилирования, то часть пакетов будет отброшена даже в том случае, если устройство не перегружено и могло бы отлично справиться с предложенным трафиком без применения механизмов QoS.

Что же меняется в сети после резервирования? При поступлении на входной интерфейс коммутатора пакетов потока, для которых было выполнено резервирование, механизм классификации распознает пакеты, относящиеся к этому потоку, и направляет их в нужную очередь. При этом пакеты могут проходить через механизм профилирования, призванный предотвратить ситуацию обслуживания потока, скорость которого превышает оговоренную при резервировании.

В результате резервирования сеть оказывается загруженной рационально. В ней нет ресурсов, которые работают со значительной перегрузкой. Механизмы организации очередей по-прежнему обеспечивают временную буферизацию пакетов в периоды пульсаций. Так как мы планировали загрузку ресурсов из расчета средних скоростей передачи данных, то на периодах пульсаций в течение некоторого ограниченного времени скорости потоков могут превышать средние скорости, так что механизмы борьбы с перегрузками по-прежнему нужны. Для обеспечения требуемых средних скоростей потоков на периодах перегрузок соответствующие потоки могут обслуживаться с помощью взвешенных очередей.

Сохраняется также главное преимущество метода коммутации пакетов: если некоторый поток не расходует отведенной ему пропускной способности, то она может выделяться для обслуживания другого потока. Нормальной практикой является резервирование пропускной способности только для части потоков, в то время как другие потоки обслуживаются без резервирования, получая обслуживание по возможности (с максимальными усилиями). Временно свободная пропускная способность может для таких потоков выделяться динамически, без нарушения взятых обязательств по обслуживанию потоков, для которых ресурсы зарезервированы.

ПРИМЕР-АНАЛОГИЯ

Проиллюстрируем принципиальное отличие резервирования ресурсов в сетях с коммутацией пакетов и каналов на примере автомобильного трафика. Пусть в некотором городе решили обеспечить некоторые привилегии для движения машин скорой помощи. В ходе обсуждения этого проекта возникли две конкурирующие идеи его реализации. Первый вариант предусматривал на всех дорогах города выделение для автомобилей скорой помощи отдельной полосы, недоступной для другого транспорта ни при каких условиях, даже если в какой-то период времени машин скорой помощи на дороге нет. Во втором случае для машин скорой помощи также выделялась отдельная полоса, но в отсутствии привилегированных машин по ней разрешалось двигаться и другому транспорту. В случае же появление машины скорой помощи автомобили, занимающие выделенную полосу, обязаны были ее освободить. Нетрудно заметить, что первый вариант соответствует принципу резервирования в сетях с коммутацией каналов — пропускная способность выделенной полосы монопольно используется автомобилями скорой помощи и не может быть перераспределена даже тогда, когда она им не нужна. Второй вариант является аналогией резервирования в сетях с коммутацией пакетов. Пропускная способность дороги здесь расходуется более эффективно, но для потока автомобилей скорой помощи такой вариант менее благоприятен, так как при необходимости освобождения полосы возникают помехи, создаваемые непривилегированными машинами.

Сеть с коммутацией каналов подобного перераспределения ресурсов выполнить не может, так как у нее в распоряжении нет независимо адресуемых единиц информации — пакетов!

Обеспечение заданного уровня задержек

При описании процедуры резервирования пропускной способности мы сфокусировались на механизмах выделения пропускной способности некоторому потоку и оставили без внимания одну важную деталь: какую пропускную способность должен запрашивать поток для того, чтобы задержки его пакетов не превышали некоторой величины? Единственное соображение, которое было высказано по этому поводу, заключалось в том, что запрашиваемая пропускная способность должна быть выше, чем максимальная скорость потока, иначе некоторая часть пакетов просто может постоянно отбрасываться сетью, так что качество обслуживания окажется гарантированно низким.

Однако эта «деталь» на самом деле оборачивается сложной проблемой, так как мы не можем, например, сконфигурировать очередь приоритетного или взвешенного обслуживания так, чтобы она строго обеспечила какой-либо заранее заданный порог задержек и их вариации. Направление пакетов в приоритетную очередь только позволяет гарантировать, что задержки будут достаточно низкими — существенно ниже, чем у пакетов, которые обрабатываются в очереди по умолчанию. Мы также знаем, что при наличии взвешенных очередей задержки будут снижаться со снижением относительного коэффициента использования пропускной способности, отведенной очереди. Но это все качественные рассуждения, а вот количественно оценить значения задержек очень сложно.

Каким же образом поставщик услуг может выполнить свои обязательства перед клиентами? Очень «просто» — он должен постоянно *измерять фактические значения характеристик трафика в сети* и гарантировать пользователям сети величины задержек в соответствии с наблюдаемыми результатами.

ПРИМЕР

Пусть сеть предоставляет три уровня качества обслуживания трафика: золотой для очень чувствительного к задержкам трафика, серебряный для трафика, чувствительного к задержкам и требующего гарантированной пропускной способности, и бронзовый для трафика, обслуживаемого по возможности. Оператор сети может различными способами добиться того, чтобы на золотом уровне обслуживания действительно гарантировались очень низкие величины задержек, вариаций задержек и потерь пакетов для трафика, на серебряном — достаточно низкие значения этих характеристик, но выше, чем у золотого, а на бронзовом гарантировались только определенные величины потерь пакетов и вовсе не гарантировались значения задержек. Для реализации такой стратегии обслуживания оператор может, например, организовать на всех коммутаторах сети приоритетную очередь для обслуживание золотого трафика и отвести ей 25 % пропускной способности на каждом выходном интерфейсе; взвешенную очередь с 50 % пропускной способности для серебряного трафика и взвешенную очередь с оставшимися 25 % для бронзового трафика. А далее он должен принимать на обслуживание потоки пользователей в каждый класс и выполнять постоянный мониторинг характеристик трафика каждого класса. И если, например, мониторинг показывает, что задержки у 95 % пакетов золотого трафика не превышают 15 мс, то оператор может гарантировать эту величину пользователям золотого уровня обслуживания. Но так как оператору нужно быть готовым к приему на обслуживание новых пользователей, то естественно было бы оставлять некоторый запас и га-

рантировать, скажем, задержку в 20 мс вместо фактического значения в 15 мс. Аналогичным образом нужно поступать с серебряным трафиком, а для бронзового достаточно измерять только долю потерь пакетов. В том случае, когда мониторинг показывает приближение фактических параметров трафика определенного уровня обслуживания к гарантируемым, можно либо добавить пропускную способность для очередей этого класса, либо прекратить прием новых пользователей в этот класс.

Как видно даже из этого краткого описания, гарантирование уровня задержек в сети является весьма сложным делом; этим объясняется тот факт, что часто операторы предпочитают давать качественное описание различных классов услуг, говоря, например, о минимальных задержках наивысшего класса обслуживания, но не давая количественных гарантий.

Инжиниринг трафика

При рассмотрении системы обеспечения качества обслуживания, основанной на резервировании, мы не стали затрагивать вопрос маршрутов следования потоков через сеть. Точнее, мы считали, что маршруты каким-то образом выбраны, причем этот выбор делается без учета требований QoS. И в условиях заданности маршрутов мы старались обеспечить прохождение по этим маршрутам такого набора потоков, для которого можно гарантировать соблюдение требований QoS.

Очевидно, что задачу обеспечения требований QoS можно решить более эффективно, если считать, что маршруты следования трафика не фиксированы, а также подлежат выбору. Это позволило бы сети обслуживать больше потоков с гарантиями QoS при тех же характеристиках самой сети, то есть пропускной способности каналов и производительности коммутаторов и маршрутизаторов.

Задачу выбора маршрутов для потоков (или классов) трафика с учетом соблюдения требований QoS решают методы **инжиниринга трафика** (Traffic Engineering, TE). С помощью этих методов стремятся добиться еще одной цели — по возможности максимально и сбалансировано загрузить все ресурсы сети, чтобы сеть при заданном уровне качества обслуживания обладала как можно более высокой суммарной производительностью.

Методы TE, как и другие рассмотренные ранее методы, основаны на резервировании ресурсов. То есть они не только позволяют найти рациональный маршрут для потока, но и резервируют для него пропускную способность ресурсов сети, находящихся вдоль этого маршрута.

Методы инжиниринга трафика являются сравнительно новыми для сетей с коммутацией пакетов. Это объясняется во многом тем, что передача эластичного трафика не предъявляла строгих требований к параметрам QoS. Кроме того, Интернет долгое время не являлся коммерческой сетью, поэтому задача максимального использования ресурсов не считалась первоочередной для IP-технологий, лежащих в основе Интернета.

Сегодня ситуация изменилась. Сети с коммутацией пакетов должны передавать различные виды трафика с заданным качеством обслуживания, максимально используя возможности своих ресурсов. Однако для этого им нужно изменить некоторые, ставшие уже традиционными, подходы к выбору маршрутов.

Недостатки традиционных методов маршрутизации

Основным принципом работы протоколов маршрутизации в сетях с коммутацией пакетов вот уже долгое время является выбор маршрута на основе топологии сети без учета информации о ее текущей загрузке.

Для каждой пары «адрес источника — адрес назначения» такие протоколы выбирают единственный маршрут, не принимая во внимание информационные потоки, протекающие через сеть. В результате все потоки между парами конечных узлов сети идут по *кратчайшему* (в соответствии с некоторой метрикой) маршруту. Выбранный маршрут может быть более рациональным, например, если в расчет принимается номинальная пропускная способность каналов связи или вносимые ими задержки, или менее рациональным, если учитывается только количество промежуточных маршрутизаторов между исходным и конечным узлами.

ВНИМАНИЕ

В традиционных методах маршрутизации наилучший выбранный маршрут рассматривается в качестве единственно возможного, даже если существуют другие, хотя и несколько худшие маршруты.

Классическим примером неэффективности такого подхода является так называемая «рыба» — сеть с топологией, приведенной на рис. 7.17. Несмотря на то что между коммутаторами *A* и *E* существует два пути (верхний — через коммутатор *B*, и нижний — через коммутаторы *C* и *D*), весь трафик от коммутатора *A* к коммутатору *E* в соответствии с традиционными принципами маршрутизации направляется по верхнему пути. Только потому, что нижний путь немного (на один ретрансляционный участок) длиннее, чем верхний, он игнорируется, хотя мог бы работать «параллельно» с верхним путем.

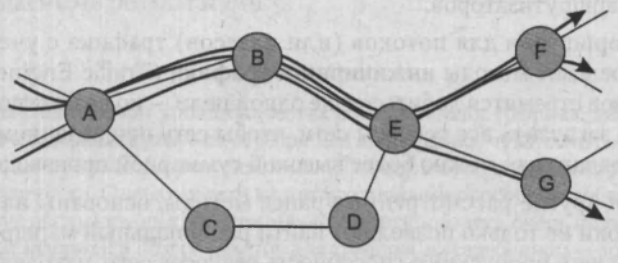


Рис. 7.17. Неэффективность кратчайших путей

Такой подход приводит к тому, что даже если кратчайший путь перегружен, пакеты все равно посылаются по этому пути. Так, в сети, представленной на рис. 7.17, верхний путь будет продолжать использоваться даже тогда, когда его ресурсов перестанет хватать для обслуживания трафика от коммутатора *A* к коммутатору *E*, а нижний путь будет простаивать, хотя, возможно, ресурсов коммутаторов *B* и *C* хватило бы для качественной передачи этого трафика.

Налицо явная ущербность методов распределения ресурсов сети — одни ресурсы работают с перегрузкой, а другие не используются вовсе. Традиционные методы борьбы с перегрузками эту проблему решить не могут, нужны качественно иные механизмы.

Методы инжиниринга трафика

Исходными данными для методов инжиниринга трафика являются:

- характеристики передающей сети — ее топология, а также производительность составляющих ее коммутаторов и линий связи (рис. 7.18);
- сведения о предложенной нагрузке сети, то есть о потоках трафика, которые сеть должна передать между своими пограничными коммутаторами (рис. 7.19).

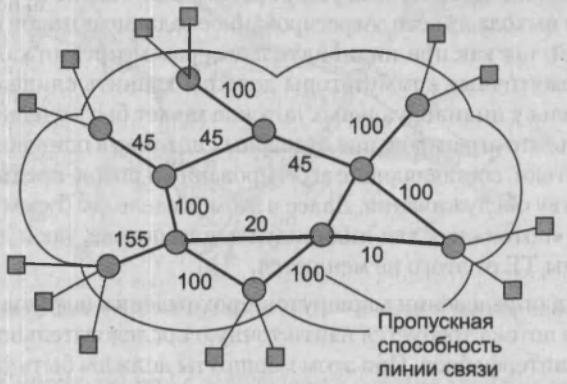


Рис. 7.18. Топология сети и производительность ее ресурсов

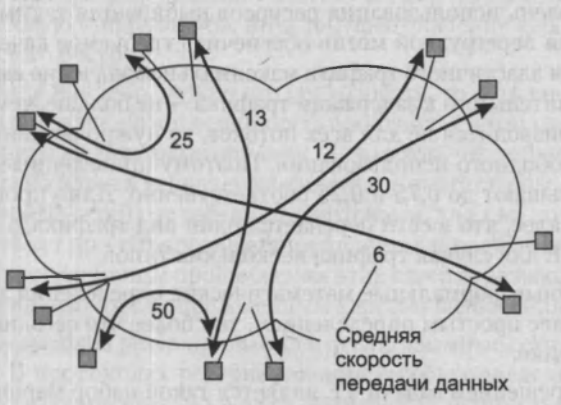


Рис. 7.19. Предложенная нагрузка

Пусть производительность процессора каждого коммутатора достаточна для обслуживания трафика всех его входных интерфейсов, даже если трафик поступает на интерфейс с максимально возможной скоростью, равной пропускной способности интерфейса. Поэтому при резервировании ресурсов будем считать ресурсами пропускную способность линий связи между коммутаторами, которая определяет также пропускную способность двух интерфейсов, связанных этой линией.

Каждый поток характеризуется точкой входа в сеть, точкой выхода из сети и профилем трафика. Для получения оптимальных решений можно использовать детальное описание каждого потока, например, учитывать величину возможной пульсации трафика или

требования QoS. Однако поскольку количественно оценить их влияние на работу сети достаточно сложно, а влияние этих параметров на характеристики QoS менее значимо, то для нахождения субоптимального распределения путей прохождения потоков через сеть, как правило, учитываются только их средние скорости передачи данных, что и показано на рис. 7.19.

Методы инжиниринга трафика чаще применяют не к отдельным, а к *агрегированным* потокам, которые являются объединением нескольких потоков. Так как мы ищем общий маршрут для нескольких потоков, то агрегировать можно только потоки, имеющих общие точки входа в сеть и выхода из сети. Агрегированное задание потоков позволяет упростить задачу выбора путей, так как при индивидуальном рассмотрении каждого пользовательского потока промежуточные коммутаторы должны хранить слишком большие объемы информации, поскольку индивидуальных потоков может быть очень много. Необходимо, однако, подчеркнуть, что агрегирование отдельных потоков в один возможно только в том случае, когда все потоки, составляющие агрегированный поток, предъявляют одни и те же требования к качеству обслуживания. Далее в этом разделе мы будем для краткости пользоваться термином «поток» как для индивидуального потока, так и для агрегированного, поскольку принципы TE от этого не меняются.

Задача TE состоит в определении маршрутов прохождения потоков трафика через сеть, то есть для каждого потока требуется найти точную последовательность промежуточных коммутаторов и их интерфейсов. При этом маршруты должны быть такими, чтобы все ресурсы сети были нагружены до максимально возможного уровня, а каждый поток получал требуемое качество обслуживания.

Максимальный уровень использования ресурсов выбирается таким образом, чтобы механизмы управления перегрузкой могли обеспечить требуемое качество обслуживания. Это означает, что для эластичного трафика максимальное значение выбирается не больше, чем 0,9, а для чувствительного к задержкам трафика — не больше, чем 0,5. Так как обычно резервирование производится не для всех потоков, то нужно оставить часть пропускной способности для свободного использования. Поэтому приведенные максимальные значения обычно уменьшают до 0,75 и 0,25 соответственно. Для упрощения рассуждений мы будем считать далее, что в сети передается один вид трафика, а потом покажем, как обобщить методы TE для случая трафика нескольких типов.

Существуют различные формальные математические определения задачи TE. Мы здесь ограничимся наиболее простым определением, тем более что сегодня оно чаще всего используется на практике.

Будем считать, что решением задачи TE является такой набор маршрутов для заданного множества потоков трафика, для которого все значения коэффициентов использования ресурсов вдоль маршрута следования каждого потока не превышают некоторого заданного порога K_{max}.

На рис. 7.20 показано одно из возможных решений задачи, иллюстрируют которую рис. 7.18 и 7.19. Найденные маршруты гарантируют, что максимальный коэффициент использования любого ресурса для любого потока не превышает 0,6.

Решение задачи TE можно искать по-разному. Во-первых, можно искать его заблаговременно, *в фоновом режиме*. Для этого нужно знать исходные данные: топологию и производительность сети, входные и выходные точки потоков трафика, среднюю скорость передачи данных в них. После этого задачу рационального распределения путей следования трафика при фиксированных точках входа и выхода, а также заданном уровне максимального

значения коэффициента использования ресурса можно передать некоторой программе, которая, например, с помощью направленного перебора вариантов найдет решение. Результатом работы программы будут точные маршруты для каждого потока с указанием всех промежуточных коммутаторов.

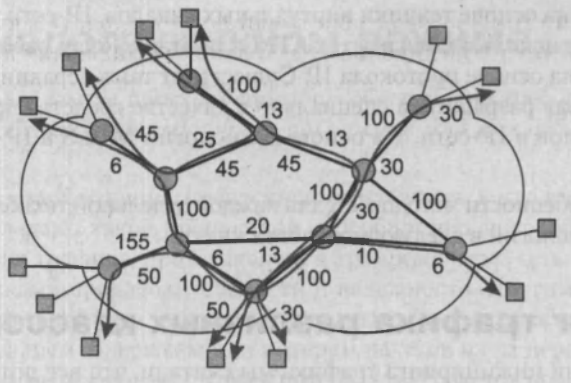


Рис. 7.20. Распределение нагрузки по сети — выбор путей передачи трафика

Во-вторых, задачу TE можно решать *в оперативном режиме*, поручив ее самим коммутаторам сети. Для этого используются модификации стандартных протоколов маршрутизации. Модификация протоколов маршрутизации состоит в том, что они сообщают друг другу не только топологическую информацию, но и текущее значение свободной пропускной способности для каждого ресурса.

После того как решение найдено, нужно его реализовать, то есть воплотить в таблицах маршрутизации. На этом этапе может возникнуть проблема — в том случае, если мы хотим проложить эти маршруты в дейтаграммной сети. Дело в том, что таблицы маршрутизации в них учитывают только адреса назначения пакетов. Коммутаторы и маршрутизаторы таких сетей (например, IP-сетей) не работают с потоками, для них поток в явном виде не существует, каждый пакет при его продвижении является независимой единицей коммутации. Можно сказать, что таблицы продвижения этих сетей отражают только топологию сети (направления продвижения к определенным адресам назначения).

Поэтому привнесение методов резервирования в дейтаграммные сети происходит с большими трудностями. В протоколах резервирования, чтобы определить поток для дейтаграммного маршрутизатора помимо адреса назначения используется некоторый дополнительный набор признаков. При этом понятие потока требуется только на этапе резервирования, а при продвижении пакетов по-прежнему работает традиционная для этого типа сетей схема, учитывающая только адрес назначения.

Теперь представим ситуацию, когда у нас имеется несколько потоков между двумя конечными узлами, и мы хотим направить их по разным маршрутам. Мы приняли такое решение, исходя из баланса загрузки сети, когда решали задачу инжиниринга трафика. Дейтаграммный коммутатор или маршрутизатор не имеет возможности реализовать наше решение, потому что для всех этих потоков у него в таблице продвижения есть только одна запись, соответствующая общему адресу назначения пакетов этих потоков. Изменять логику работы коммутаторов и маршрутизаторов дейтаграммных сетей достаточно нецелесообразно, поскольку это слишком принципиальная модернизация.

В результате методы инжиниринга трафика сегодня используются только в сетях с виртуальными каналами, для которых не составляет труда реализовать найденное решение для группы потоков. Каждому потоку (или группе потоков с одинаковыми маршрутами) выделяется виртуальный канал, который прокладывается в соответствии с выбранным маршрутом. Методы инжиниринга трафика успешно применяются в сетях ATM и Frame Relay, работающих на основе техники виртуальных каналов. IP-сети также опираются на методы TE, когда те используются в сетях ATM или Frame Relay, работающих в составной сети, построенной на основе протокола IP. Существует также сравнительно новая технология MPLS, которая разработана специально в качестве средства привнесения техники виртуальных каналов в IP-сети. На основе технологии MPLS в IP-сетях можно также решать задачи TE.

Мы рассмотрим особенности методов TE для каждой отдельной технологии при детальном изучении этих технологий в следующих частях книги.

Инжиниринг трафика различных классов

При решении задачи инжиниринга трафика мы считали, что все потоки трафика предъявляли одинаковые требования к качеству обслуживания. То есть пользователей сети удовлетворяло, что все потоки обслуживаются с заданной средней скоростью (она, естественно, у каждого потока своя, отличающаяся от других).

Более реальной является ситуация, когда у каждого пользователя сети имеется *несколько классов трафика*, и эти классы отличаются разными требованиями к качеству обслуживания. Мы уже обсуждали эту проблему при рассмотрении вопросов резервирования ресурсов.

В методах TE, учитывающих наличие в сети трафика с различными требованиями QoS, проблема решается точно так же, как и в методах резервирования ресурсов отдельных узлов. Если у нас имеется, например, два класса трафика, то мы задаемся двумя уровнями максимального использования ресурсов.

Для достижения такого результата с каждым ресурсом должно быть связано два счетчика свободной пропускной способности — один для приоритетного, второй для эластичного трафика. При определении возможности прохождения маршрута через конкретный ресурс для приоритетного трафика средняя интенсивность нового потока должна сравниваться со свободной пропускной способностью для приоритетного трафика.

Если свободной пропускной способности достаточно и новый поток пойдет через данный интерфейс, то значение средней скорости передачи данных для нового потока необходимо вычесть как из счетчика загрузки приоритетного трафика, так и из счетчика загрузки эластичного трафика, так как приоритетный трафик всегда будет обслуживаться перед эластичным и создаст для эластичного трафика дополнительную нагрузку Если же задача TE решается для эластичного трафика, то его средняя скорость передачи данных сравнивается со свободной пропускной способностью счетчика эластичного трафика и в случае положительного решения значение этой скорости вычитается только из счетчика эластичного трафика, так как для приоритетного трафика эластичный трафик прозрачен.

Модифицированные протоколы маршрутизации должны распространять по сети информацию о двух параметрах свободной пропускной способности — для каждого класса трафика отдельно. Если же задача обобщается для случая передачи через сеть трафика нескольких

классов, то, соответственно, с каждым ресурсом должно быть связано *столько счетчиков, сколько классов трафика существует в сети*, а протоколы маршрутизации должны распространять вектор свободных пропускных способностей соответствующей размерности.

Работа в недогруженном режиме

Как мы уже отмечали, самым простым способом обеспечения требований QoS для всех потоков является *работа сети в недогруженном режиме*, или же *с избыточной пропускной способностью*.

Говорят, что сеть имеет избыточную пропускную способность, когда все части сети в любой момент времени обладают такой пропускной способностью, которой достаточно, чтобы обслужить все потоки трафика, протекающего в это время через сеть, с удовлетворительными характеристиками производительности и надежности. Другими словами, ни одно из сетевых устройств такой сети никогда не подвергается перегрузкам, которые могли бы привести к значительным задержкам или потерям пакетов из-за переполнения очередей пакетов (конечно, это не исключает случаев потерь сетью пакетов по другим причинам, не связанным с перегрузкой сети, например, из-за искажений сигналов на линиях связи либо отказов сетевых узлов или линий связи).

Простота этого подхода является его главным достоинством, так как он требует только увеличения пропускной способности линий связи и, соответственно, производительности коммуникационных устройств сети. Никаких дополнительных усилий по исследованию характеристик потоков сети и конфигурированию дополнительных очередей и механизмов кондиционирования трафика, как в случае применения методов QoS, здесь не требуется.

Заметим, что определение сети с избыточной пропускной способностью было намеренно упрощено, чтобы передать суть идеи. Более точное определение должно учитывать случайный характер протекающих в сети процессов и оперировать статистическими определениями событий, то есть говорить, что такие события, как длительные задержки или потери пакетов из-за переполнения очередей в сети с избыточной пропускной способностью, случаются так редко, что ими можно пренебречь. В результате трафик всех приложений в подобной сети переносится с высоким качеством.

Однако доказать, что сеть действительно является сетью с избыточной пропускной способностью, на практике достаточно трудно. Только постоянное измерение времен доставки пакетов всем конечным узлам сети может показать, что сеть удовлетворяет данному описанию — мы уже сталкивались с этой ситуацией, когда рассматривали механизм гарантирования определенного уровня задержек пакетов при применении методов QoS.

Однако мониторинг задержек и их вариаций является тонкой и трудоемкой работой. Обычно операторы, которые хотят поддерживать свою сеть в недогруженном состоянии и за счет этого обеспечивать высокое качество обслуживания, поступают проще — они осуществляют *мониторинг уровня трафика в линиях связи сети*, то есть *измеряют коэффициент использования пропускной способности линий связи*. При этом линия связи считается недогруженной, если ее коэффициент использования постоянно не превосходит некоторый достаточно низкий уровень, например 10 %. Имея такие значения измерений, можно считать, что линия в среднем не испытывает перегрузок, а значит, задержки пакетов

будут низкими — мы знаем о такой зависимости между коэффициентом загрузки ресурса и задержками из теории массового обслуживания, рассмотренной на примере простейшей модели M/M/1.

Однако даже столь низкие значения загрузки не исключают появления на линии кратковременных пульсаций трафика, способных приводить к повышению пиковой скорости трафика до величины пропускной способности линии и, следовательно, к значительным задержкам или потерям небольшого количества пакетов. Для некоторых типов приложений такие потери могут быть весьма чувствительными.

Многие средства мониторинга скорости трафика, особенно встроенные в коммутаторы и маршрутизаторы, измеряют скорость трафика, усредняя ее на слишком длинных интервалах. В результате такие средства мониторинга просто не способны зарегистрировать кратковременные пульсации трафика и часто дают слишком оптимистичную оценку загруженности сети.

Эту проблему иллюстрирует рис. 7.21. На нем показаны результаты измерения скорости трафика на интерфейсе с пропускной способностью в 2 Мбит/с.

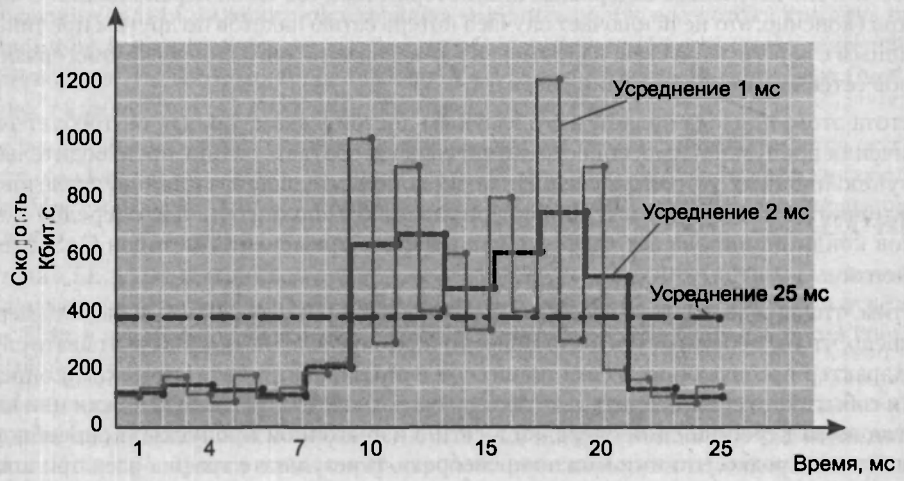


Рис. 7.21. Зависимость результатов измерений скорости трафика от времени усреднения

На рисунке представлены три кривые, полученные для одного и того же трафика при различных интервалах усреднения данных. Серой сплошной линией показаны результаты, полученные для интервала усреднения данных в 1 мс; пунктирная черная линия демонстрирует результаты для интервала усреднения в 2 мс, а штрих-пунктирная черная линия соответствует интервалу в 25 мс.

Обычная практика для оценки состояния недогруженности интерфейса состоит в использовании предела в 25 % от его пропускной способности как индикатора недогруженности. Для нашего примера это соответствует скорости трафика 500 Кбит/с.

Тогда, используя результаты мониторинга интерфейса с интервалом усреднения в 25 мс, мы уверенно считаем, что интерфейс недогружен и нам не стоит беспокоиться о возможных задержках и потерях пакетов из-за перегрузок интерфейса. Однако глядя на серую кривую (усреднение 1 мс), мы видим, что в шести интервалах скорость намного превышала

500 Кбит/с, а значит, на этих интервалах длительные задержки и потери пакетов вполне могли случиться. Наконец, данные, полученные при усреднении в 2 мс, показывают, что интерфейс находится вблизи границы недогруженности.

Данные, использованные для построения кривых на рис. 7.21, были искусственно подобраны так, чтобы показать крайние ситуации. Однако эти кривые действительно отражают тонкий и важный эффект измерений, который нужно учитывать при мониторинге загрузки линий связи сети: слишком длительные интервалы усреднения при измерении скорости могут существенно исказить картину и привести к потере важной информации, а в конечном итоге — к переоценке возможностей сети качественно передавать трафик. Часто на практике выполняют мониторинг загрузки линий связи с 5-секундным интервалом усреднения, что явно недостаточно для оценки состояния сети.

Для более достоверной оценки состояния сети нужно дополнять мониторинг загрузки линий связи сети хотя бы выборочным мониторингом характеристик QoS, таких как задержки, вариации задержек и потери пакетов. В этом случае можно с большей уверенностью говорить о том, что сеть действительно является сетью с избыточной пропускной способностью, которая гарантирует всем типам трафика качественное обслуживание. Кроме того, выборочный мониторинг характеристик QoS может помочь в определении предела загрузки линий, служащего для оценки их недогруженности. В нашем примере в качестве такого предела мы использовали значение 25 %, но вполне возможно, что это эмпирическое значение для некоторой конкретной сети требуется уточнить.

Выводы

Качество обслуживания в его узком смысле фокусирует внимание на характеристиках и методах передачи трафика через очереди коммуникационных устройств. Методы обеспечения качества обслуживания занимают сегодня важное место в семействе технологий сетей с коммутацией пакетов, так как без их применения сложно обеспечить качественную работу современных мультимедийных приложений, таких как IP-телефония, видео- и радиовещание, интерактивное дистанционное обучение и т. п.

Характеристики QoS отражают отрицательные последствия пребывания пакетов в очередях, которые проявляются в снижении скорости передачи, задержках пакетов и их потерях.

Существуют различные типы трафика, отличающиеся чувствительностью к задержкам и потерям пакетов. Наиболее грубая классификация трафика разделяет его на два класса: трафик реального времени (чувствительный к задержкам) и эластичный трафик (нечувствительный к задержкам в широких пределах).

Методы QoS основаны на перераспределении имеющейся пропускной способности линий связи между трафиком различного типа в соответствии с требованиями приложений.

Приоритетные и взвешенные очереди являются основным инструментом выделения пропускной способности определенным потокам пакетов.

Механизм профилирования позволяет контролировать скорость потока пакетов и ограничивать ее в соответствии с заранее заданным уровнем.

Обратная связь является одним из механизмов QoS; она позволяет временно снизить скорость поступления пакетов в сеть для ликвидации перегрузки в узле сети.

Резервирование пропускной способности «из конца в конец» позволяет добиться гарантированного качества обслуживания для потока пакетов. Резервирование основано на процедуре контроля допуска потока в сеть, в ходе которой проверяется наличие доступной пропускной способности для обслуживания потока вдоль маршрута его следования.

Методы инжиниринга трафика состоят в выборе рациональных маршрутов прохождения потоков через сеть. Выбор маршрутов обеспечивает максимизацию загрузки ресурсов сети при одновременном соблюдении необходимых гарантий в отношении параметров качества обслуживания трафика.

Недогруженная сеть (она же сеть с избыточной пропускной способностью) может обеспечить качественное обслуживание трафика всех типов без применения методов QoS; однако для того чтобы убедиться, что сеть действительно недогружена, требуется постоянно проводить мониторинг уровней загрузки линий связи сети, выполняя измерения с достаточно высокой частотой.

Вопросы и задания

1. В чем причина возникновения очередей в сетях с коммутацией пакетов? Возникают ли очереди в сетях с коммутацией каналов?
2. Какой параметр в наибольшей степени влияет на размер очереди?
3. К каким нежелательным последствиям может привести приоритетное обслуживание?
4. На какие два класса можно разделить приложения в отношении предсказуемости скорости передачи данных?
5. При увеличении пульсации некоторого потока увеличатся или уменьшатся задержки, связанные с пребыванием пакетов этого потока в очереди (при сохранении всех других параметров потока и условий его обслуживания)?
6. Какому элементу коммутатора или маршрутизатора чаще всего соответствует обслуживающий прибор модели M/M/1?
7. Объясните причину возможного возникновения очередей даже при невысокой средней загрузке коммутаторов или маршрутизаторов сети с коммутацией пакетов?
8. Для трафика какого типа в наибольшей степени подходит взвешенное обслуживание? Варианты ответов:

   а) трафика видеоконференций;

   б) трафика загрузки больших файлов данных;

   в) трафика IP-телефонии.

   А приоритетное обслуживание?
9. Можно ли комбинировать приоритетное и взвешенное обслуживание?
10. Какой из трех потоков будет меньше в среднем задерживаться в очереди к выходному интерфейсу 100 Мбит/с, если потоки обслуживаются взвешенными очередями, при этом потокам отведено 60, 30 и 10 % пропускной способности интерфейса соответственно? Потоки имеют средние скорости: 50, 15 и 7 Мбит/с соответственно. Коэффициент вариации интервалов следования пакетов одинаков у всех потоков.
11. Что является причиной того, что поток, который обслуживается в очереди самого высокого приоритета, все равно сталкивается с необходимостью ожидания в очереди? Варианты ответов:

    а) очереди более низких приоритетов;

    б) собственная пульсация;

    в) пульсации низкоприоритетного трафика.

12. Может ли пропускная способность, зарезервированная в сети с коммутацией пакетов для потока A, использоваться потоком B?
13. Какой параметр трафика меняется при инжиниринге трафика?
14. Почему обычные протоколы маршрутизации не используются при решении задач инжиниринга трафика? Варианты ответов:
    а) они не обеспечивают быстрого нахождения нового маршрута при отказах элементов сети;
    б) они не позволяют прокладывать различные маршруты для потоков с одним и тем же адресом назначения;
    в) при выборе маршрута они не учитывают свободной пропускной способности линий связи сети.
15. Каковы преимущества и недостатки метода работы сети в недогруженном режиме по сравнению с методами QoS?
16. Мониторинг какой характеристики сети обычно выполняют операторы связи при работе сети в недогруженном режиме без применения механизмов QoS?

Часть II

Технологии физического уровня

Физической основой любой компьютерной (и телекоммуникационной) сети являются линии связи. Без таких линий коммутаторы не могли бы обмениваться пакетами, и компьютеры оставались бы изолированными устройствами.

После изучения принципов построения компьютерных сетей в воображении читателя могла возникнуть достаточно простая картина компьютерной сети — компьютеры и коммутаторы, соединенные друг с другом отрезками кабеля. Однако при более детальном рассмотрении компьютерной сети все оказывается сложнее, чем это казалось при изучении модели OSI.

Дело в том, что специально выделенные кабели используются для соединения сетевых устройств только на небольших расстояниях, то есть в локальных сетях. При построении сетей WAN и MAN такой подход крайне расточителен из-за высокой стоимости протяженных линий связи. К тому же на их прокладку необходимо получать разрешение. Поэтому гораздо чаще для связи коммутаторов в сетях WAN и MAN применяются уже существующие телефонные или первичные территориальные сети с коммутацией каналов. В этом случае в сети с коммутацией каналов создается составной канал, который выполняет те же функции, что и отрезок кабеля — обеспечивает физическое двухточечное соединение. Конечно, составной канал представляет собой гораздо более сложную техническую систему, чем кабель, но для компьютерной сети эти сложности прозрачны. Первичные сети специально строятся для создания канальной инфраструктуры, поэтому их каналы более эффективны по соотношению цена/пропускная способность. Сегодня в распоряжении проектировщика компьютерной сети имеются каналы первичных сетей для широкого диапазона скоростей — от 64 Кбит/с до 10 Гбит/с.

Несмотря на различия в физической и технической природе линий связи, их можно описать с помощью единого набора характеристик. Важнейшими характеристиками любой линии связи при передаче дискретной информации являются полоса пропускания, измеряемая в герцах (Гц), и емкость, или пропускная способность, измеряемая в битах в секунду (бит/с). Пропускная способность представляет собой скорость битового потока, передаваемого линией связи. Пропускная способность зависит от полосы пропускания линии и способа кодирования дискретной информации.

Все большую популярность приобретают беспроводные каналы. Они являются единственным типом каналов, обеспечивающих мобильность пользователей компьютерной сети. Кроме того, беспроводная связь применяется в тех случаях, когда кабели проложить невозможно или невыгодно — в малонаселенных районах, при доступе к жилым домам, уже охваченным кабельной инфраструктурой конкурентов и т. п. При беспроводной связи используются электромагнитные волны различной частоты — радиоволны, микроволны, инфракрасное излучение и видимый свет. Высокий уровень помех и сложные пути распространения волн требуют применения в беспроводных каналах особых способов кодирования и передачи сигналов.

- Глава 8. Линии связи
- Глава 9. Кодирование и мультиплексирование данных
- Глава 10. Беспроводная передача данных
- Глава 11. Первичные сети

ГЛАВА 8 Линии связи

При построении сетей применяются линии связи, в которых используются различные физические среды: подвешенные в воздухе телефонные и телеграфные провода, проложенные под землей и по дну океана медные коаксиальные и волоконно-оптические кабели, опутывающие все современные офисы медные витые пары, всепроникающие радиоволны.

В этой главе рассматриваются общие характеристики линий связи, не зависящие от их физической природы, такие как полоса пропускания, пропускная способность, помехоустойчивость и достоверность передачи. Ширина полосы пропускания является фундаментальной характеристикой канала связи, так как определяет максимально возможную информационную скорость канала, которая называется пропускной способностью канала. Формула Найквиста выражает эту зависимость для идеального канала, а формула Шеннона учитывает наличие в реальном канале шума. Завершает главу рассмотрение конструкций и стандартов современных кабелей, которые составляют основу проводных линий связи.

Классификация линий связи

Первичные сети, линии и каналы связи

При описании технической системы, которая передает информацию между узлами сети, в литературе можно встретить несколько названий: *линия связи*, *составной канал*, *канал*, *звено*. Часто эти термины используются как синонимы, и во многих случаях это не вызывает проблем. В то же время есть и специфика в их употреблении.

- **Звено** (link) — это сегмент, обеспечивающий передачу данных между двумя соседними узлами сети. То есть звено не содержит промежуточных устройств коммутации и мультиплексирования.
- **Каналом** (channel) чаще всего обозначают часть пропускной способности звена, используемую независимо при коммутации. Например, звено первичной сети может состоять из 30 каналов, каждый из которых обладает пропускной способностью 64 Кбит/с.
- **Составной канал** (circuit) — это путь между двумя конечными узлами сети. Составной канал образуется отдельными каналами промежуточных звеньев и внутренними соединениями в коммутаторах. Часто эпитет «составной» опускается, и термином «канал» называют как составной канал, так и канал между соседними узлами, то есть в пределах звена.
- **Линия связи** может использоваться как синоним для любого из трех остальных терминов.

Не стоит относиться к путанице в терминологии очень строго. Особенно это относится к различиям в терминологии традиционной телефонии и более новой области — компьютерных сетей. Процесс конвергенции только усугубил проблему терминологии, так как многие механизмы этих сетей стали общими, но сохранили за собой по паре (иногда и больше) названий, пришедших из каждой области.

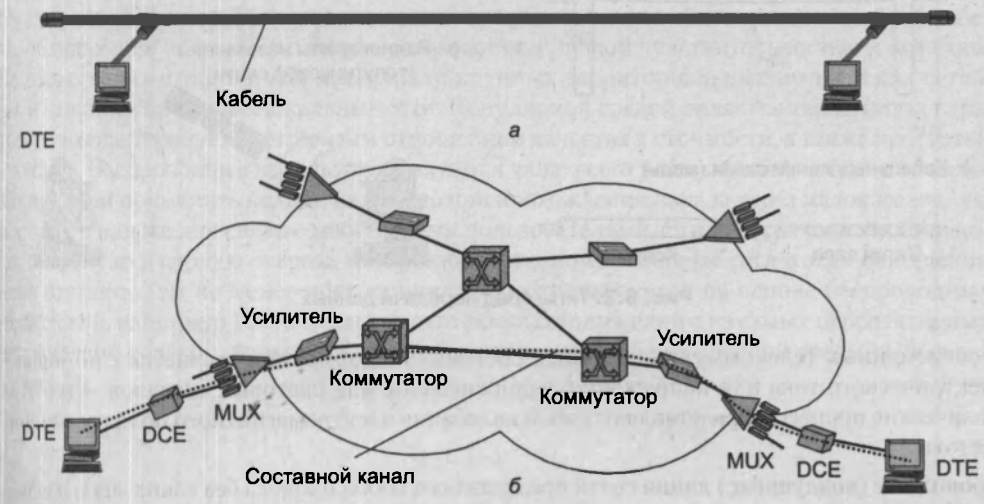


Рис. 8.1. Состав линии связи

Кроме того, существуют объективные причины для неоднозначного понимания терминов. На рис. 8.1 показаны два варианта линии связи. В первом случае (рис. 8.1, *а*) линия состоит из сегмента кабеля длиной несколько десятков метров и представляет собой звено. Во втором случае (рис. 8.1, *б*) линия связи представляет собой составной канал, развернутый в сети с коммутацией каналов. Такой сетью может быть **первичная сеть** или телефонная сеть.

Однако для компьютерной сети эта линия представляет собой звено, так как соединяет два соседних узла, и вся коммутационная промежуточная аппаратура является прозрачной для этих узлов. Повод для взаимного непонимания на уровне терминов компьютерных специалистов и специалистов первичных сетей здесь очевиден.

Первичные сети специально создаются для того, чтобы предоставлять услуги каналов передачи данных для компьютерных и телефонных сетей, про которые в таких случаях говорят, что они работают «поверх» первичных сетей и являются **наложенными сетями**.

Физическая среда передачи данных

Линии связи отличаются также физической средой, используемой для передачи информации.

Физическая среда передачи данных может представлять собой набор проводников, по которым передаются сигналы. На основе таких проводников строятся проводные (воздушные) или кабельные линии связи (рис. 8.2). В качестве среды также используется земная атмосфера или космическое пространство, через которое распространяются информационные сигналы. В первом случае говорят о *проводной среде*, а во втором — о *беспроводной*.

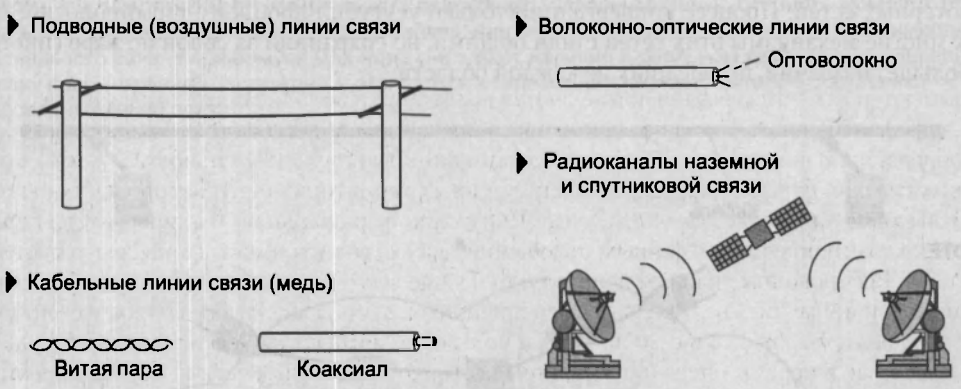


Рис. 8.2. Типы сред передачи данных

В современных телекоммуникационных системах информация передается с помощью электрического тока или напряжения, радиосигналов или световых сигналов — все эти физические процессы представляют собой колебания электромагнитного поля различной частоты.

Проводные (воздушные) линии связи представляют собой провода без каких-либо изолирующих или экранирующих оплеток, проложенные между столбами и висящие в воздухе. Еще в недалеком прошлом такие линии связи были основными для передачи телефонных

или телеграфных сигналов. Сегодня проводные линии связи быстро вытесняются кабельными. Но кое-где они все еще сохранились и при отсутствии других возможностей продолжают использоваться, в частности, и для передачи компьютерных данных. Скоростные качества и помехозащищенность этих линий оставляют желать много лучшего.

Кабельные линии имеют достаточно сложную конструкцию. Кабель состоит из проводников, заключенных в несколько слоев изоляции: электрической, электромагнитной, механической и, возможно, климатической. Кроме того, кабель может быть оснащен разъемами, позволяющими быстро выполнять присоединение к нему различного оборудования. В компьютерных (и телекоммуникационных) сетях применяются три основных типа кабеля: кабели на основе скрученных пар медных проводов — **неэкранированная витая пара** (Unshielded Twisted Pair, UTP) и **экранированная витая пара** (Shielded Twisted Pair, STP), **коаксиальные кабели** с медной жилой, **волоконно-оптические кабели**. Первые два типа кабелей называют также **медными кабелями**.

Радиоканалы наземной и спутниковой связи образуются с помощью передатчика и приемника радиоволн. Существует большое разнообразие типов радиоканалов, отличающихся как используемым частотным диапазоном, так и дальностью канала. **Диапазоны широковещательного радио** (длинных, средних и коротких волн), называемые также **АМ-диапазонами**, или диапазонами амплитудной модуляции (Amplitude Modulation, AM), обеспечивают дальнюю связь, но при невысокой скорости передачи данных. Более скоростными являются каналы, использующие **диапазоны очень высоких частот** (Very High Frequency, VHF), для которых применяется частотная модуляция (Frequency Modulation, FM). Для передачи данных также используются **диапазоны ультравысоких частот** (Ultra High Frequency, UHF), называемые еще **диапазонами микроволн** (свыше 300 МГц). При частоте свыше 30 МГц сигналы уже не отражаются ионосферой Земли, и для устойчивой связи требуется наличие прямой видимости между передатчиком и приемником. Поэтому указанные частоты используются в спутниковых или радиорелейных каналах либо в таких локальных или мобильных сетях, в которых это условие выполняется.

В компьютерных сетях сегодня применяются практически все описанные типы физических сред передачи данных. Хорошие возможности предоставляют волоконно-оптические кабели, обладающие широкой полосой пропускания и низкой чувствительностью к помехам. На них сегодня строятся как магистрали крупных территориальных и городских сетей, так и высокоскоростные локальные сети. Популярной средой является также витая пара, которая характеризуется отличным отношением качества к стоимости, а также простотой монтажа. Беспроводные каналы используются чаще всего в тех случаях, когда кабельные линии связи применить нельзя, например при прохождении канала через малонаселенную местность или же для связи с мобильными пользователями сети. Обеспечение мобильности затронуло в первую очередь телефонные сети, компьютерные сети в этом отношении пока отстают. Тем не менее построение компьютерных сетей на основе беспроводных технологий, например Radio Ethernet, считаются сегодня одним из самых перспективных направлений телекоммуникаций. Линии связи на основе беспроводной среды изучаются в главе 10.

Аппаратура передачи данных

Как показано на рис. 8.1, линии связи состоят не только из среды передачи, но и аппаратуры. Даже в том случае, когда линия связи не проходит через первичную сеть, а основана на кабеле, в ее состав входит аппаратура передачи данных.

Аппаратура передачи данных (Data Circuit Equipment, DCE) в компьютерных сетях непосредственно присоединяет компьютеры или коммутаторы к линиям связи и является, таким образом, пограничным оборудованием. Традиционно аппаратуру передачи данных включают в состав линии связи. Примерами DCE являются **модемы** (для телефонных линий), **терминальные адаптеры сетей ISDN**, **устройства для подключения к цифровым каналам** первичных сетей DSU/CSU (Data Service Unit/Circuit Service Unit).

DCE работает на физическом уровне модели OSI, отвечая за передачу информации в физическую среду (в линию) и прием из нее сигналов нужной формы, мощности и частоты. Аппаратура пользователя линии связи, вырабатывающая данные для передачи по линии связи и подключаемая непосредственно к аппаратуре передачи данных, носит обобщенное название **оконечное оборудование данных** (Data Terminal Equipment, DTE). Примером DTE могут служить компьютеры, коммутаторы и маршрутизаторы. Эту аппаратуру не включают в состав линии связи.

ПРИМЕЧАНИЕ

Разделение оборудования на DCE и DTE в локальных сетях является достаточно условным. Например, адаптер локальной сети можно считать как принадлежностью компьютера, то есть оборудованием DTE, так и составной частью канала связи, то есть аппаратурой DCE. Точнее, одна часть сетевого адаптера выполняет функции DTE, а его другая, оконечная его часть, непосредственно принимающая и передающая сигналы, относится к DCE.

Для подключения DCE-устройств к DTE-устройствам (то есть к компьютерам или коммутаторам/маршрутизаторам) существует несколько *стандартных интерфейсов*[1]. Работают эти устройства на коротких расстояниях друг от друга, как правило, несколько метров.

Промежуточная аппаратура обычно используется на линиях связи большой протяженности. Она решает две основные задачи:

- улучшение качества сигнала;
- создание постоянного составного канала связи между двумя абонентами сети.

В локальных сетях промежуточная аппаратура может совсем не использоваться, если протяженность физической среды — кабелей или радиоэфира — позволяет одному сетевому адаптеру принимать сигналы непосредственно от другого сетевого адаптера без дополнительного усиления. В противном случае применяется промежуточная аппаратура, роль которой здесь играют устройства типа **повторителей** и **концентраторов**.

В *глобальных сетях* необходимо обеспечить качественную передачу сигналов на расстояния в сотни и тысячи километров. Поэтому без **усилителей** (повышающих мощность сигналов) и **регенераторов** (наряду с повышением мощности восстанавливающих форму импульс-

[1] Интерфейсы DTE-DCE описываются стандартами серии V CCITT, а также стандартами EIA серии RS (Recommended Standards — рекомендуемые стандарты). Две линии стандартов во многом дублируют друг друга. Наиболее популярными стандартами являются RS-232, RS-530, V.35 и HSSI.

ных сигналов, исказившихся при передаче на большое расстояние), установленных через определенные расстояния, построить территориальную линию связи невозможно.

В первичных сетях помимо упомянутого оборудования, обеспечивающего качественную передачу сигналов, необходима промежуточная коммутационная аппаратура — **мультиплексоры** (MUX), **демультиплексоры** и **коммутаторы**. Эта аппаратура создает между двумя абонентами сети постоянный составной канал из отрезков физической среды — кабелей с усилителями.

В зависимости от типа промежуточной аппаратуры все линии связи делятся на аналоговые и цифровые. В **аналоговых линиях** промежуточная аппаратура предназначена для усиления аналоговых сигналов, то есть сигналов, которые имеют непрерывный диапазон значений. Такие линии связи традиционно применялись в телефонных сетях с целью связи телефонных коммутаторов между собой. Для создания высокоскоростных каналов, которые мультиплексируют несколько низкоскоростных аналоговых абонентских каналов, при аналоговом подходе обычно используется техника *частотного мультиплексирования* (Frequency Division Multiplexing, FDM).

В **цифровых линиях** связи передаваемые сигналы имеют конечное число состояний. Как правило, элементарный сигнал, то есть сигнал, передаваемый за один такт работы передающей аппаратуры, имеет 2, 3 или 4 состояния, которые в линиях связи воспроизводятся импульсами или потенциалами прямоугольной формы. С помощью таких сигналов передаются как компьютерные данные, так и оцифрованные речь и изображение (именно благодаря одинаковому способу представления информации современными компьютерными, телефонными и телевизионными сетями стало возможным появление общих для всех первичных сетей). В цифровых линиях связи используется специальная промежуточная аппаратура — регенераторы, которые улучшают форму импульсов и восстанавливают период их следования. Промежуточная аппаратура мультиплексирования и коммутации первичных сетей работает по принципу *временного мультиплексирования каналов* (Time Division Multiplexing, TDM).

Характеристики линий связи

Спектральный анализ сигналов на линиях связи

Важная роль при определении параметров линий связи отводится спектральному разложению передаваемого по этой линии сигнала. Из теории гармонического анализа известно, что *любой периодический процесс можно представить в виде суммы синусоидальных колебаний различных частот и различных амплитуд* (рис. 8.3).

Каждая составляющая синусоида называется также **гармоникой**, а набор всех гармоник называют **спектральным разложением**, или **спектром**, исходного сигнала. Под **шириной спектра сигнала** понимается разность между максимальной и минимальной частотами того набора синусоид, которые в сумме дают исходный сигнал.

Непериодические сигналы можно представить в виде интеграла синусоидальных сигналов с непрерывным спектром частот. В частности, спектральное разложение идеального импульса (единичной мощности и нулевой длительности) имеет составляющие всего спектра частот, от $-\infty$ до $+\infty$ (рис. 8.4).

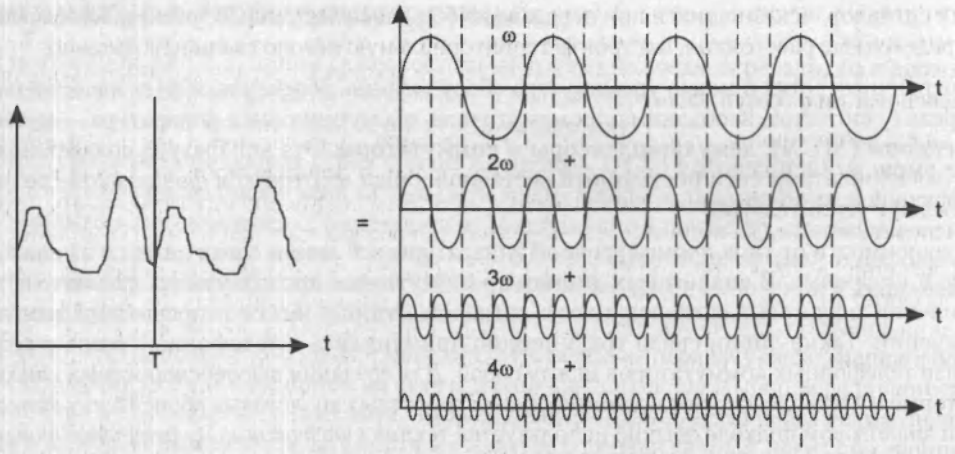


Рис. 8.3. Представление периодического сигнала суммой синусоид

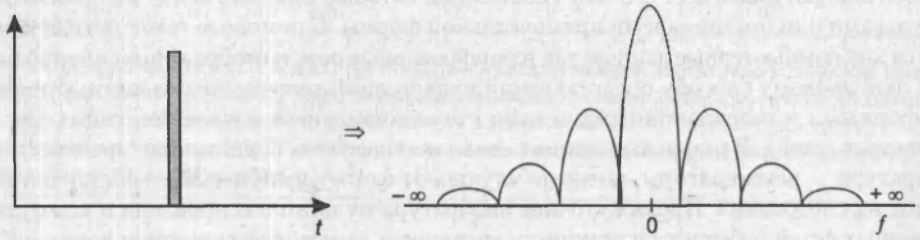


Рис. 8.4. Спектральное разложение идеального импульса

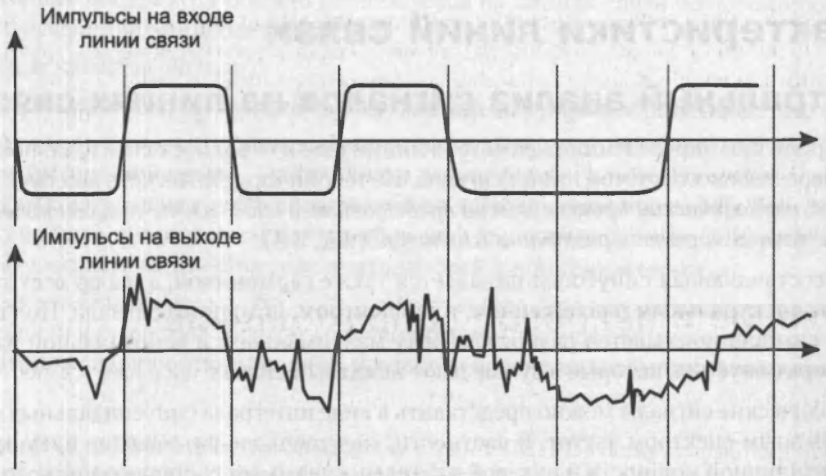


Рис. 8.5. Искажение импульсов в линии связи

Техника нахождения спектра любого исходного сигнала хорошо известна. Для некоторых сигналов, которые описываются аналитически (например, для последовательности прямоугольных импульсов одинаковой длительности и амплитуды), спектр легко вычисляется на основании **формул Фурье**.

Для сигналов произвольной формы, встречающихся на практике, спектр можно найти с помощью специальных приборов — спектральных анализаторов, которые измеряют спектр реального сигнала и отображают амплитуды составляющих гармоник на экране, распечатывают их на принтере или передают для обработки и хранения в компьютер.

Искажение передающей линией связи синусоиды какой-либо частоты приводит, в конечном счете, к искажению амплитуды и формы передаваемого сигнала любого вида. Искажения формы проявляются в том случае, когда синусоиды различных частот искажаются неодинаково. Если это аналоговый сигнал, передающий речь, то изменяется тембр голоса за счет искажения обертонов — боковых частот. При передаче импульсных сигналов, характерных для компьютерных сетей, искажаются низкочастотные и высокочастотные гармоники, в результате фронты импульсов теряют свою прямоугольную форму (рис. 8.5), и сигналы могут плохо распознаваться на приемном конце линии.

Передаваемые сигналы искажаются из-за несовершенства линий связи. Идеальная передающая среда, не вносящая никаких помех в передаваемый сигнал, должна, по меньшей мере, иметь нулевые значения сопротивления, емкости и индуктивности. Однако на практике медные провода, например, всегда представляют собой некоторую распределенную по длине комбинацию активного сопротивления, емкостной и индуктивной нагрузок (рис. 8.6). В результате синусоиды различных частот передаются этими линиями по-разному.

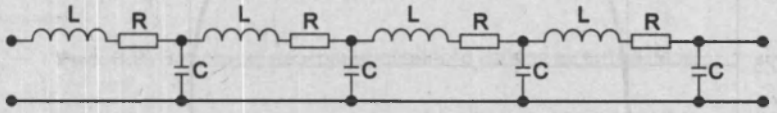


Рис. 8.6. Представление линии как распределенной индуктивно-емкостной нагрузки

Помимо искажений сигналов, возникающих из-за не идеальных физических параметров линии связи, существуют и **внешние помехи**, которые вносят свой вклад в искажение формы сигналов на выходе линии. Эти помехи создаются различными электрическими двигателями, электронными устройствами, атмосферными явлениями и т. д. Несмотря на защитные меры, предпринимаемые разработчиками кабелей, и наличие усилительной и коммутирующей аппаратуры, полностью компенсировать влияние внешних помех не удается. Помимо внешних помех в кабеле существуют и **внутренние помехи** — так называемые **наводки** одной пары проводников на другую. В результате сигналы на выходе линии связи могут иметь искаженную форму (как это и показано на рис. 8.5).

Затухание и волновое сопротивление

Степень искажения синусоидальных сигналов линиями связи оценивается такими характеристиками, как затухание и полоса пропускания.

Затухание показывает, насколько уменьшается мощность эталонного синусоидального сигнала на выходе линии связи по отношению к мощности сигнала на входе этой линии. Затухание (A) обычно измеряется в децибелах (дБ) и вычисляется по следующей формуле:

$$A = 10 \lg P_{out}/P_{in}.$$

Здесь P_{out} — мощность сигнала на выходе линии, P_{in} — мощность сигнала на входе линии. Так как затухание зависит от длины линии связи, то в качестве характеристики линии связи используется так называемое **погонное затухание**, то есть затухание на линии связи определенной длины. Для кабелей локальных сетей в качестве такой длины обычно используют 100 м, так как это значение является максимальной длиной кабеля для многих технологий LAN. Для территориальных линий связи погонное затухание измеряют для расстояния в 1 км.

Обычно затуханием характеризуют пассивные участки линии связи, состоящие из кабелей и кроссовых секций, без усилителей и регенераторов. Так как мощность выходного сигнала кабеля без промежуточных усилителей меньше, чем мощность входного, затухание кабеля всегда является *отрицательной величиной*.

Степень затухания мощности синусоидального сигнала зависит от частоты синусоиды, и эта зависимость также характеризует линию связи (рис. 8.7).

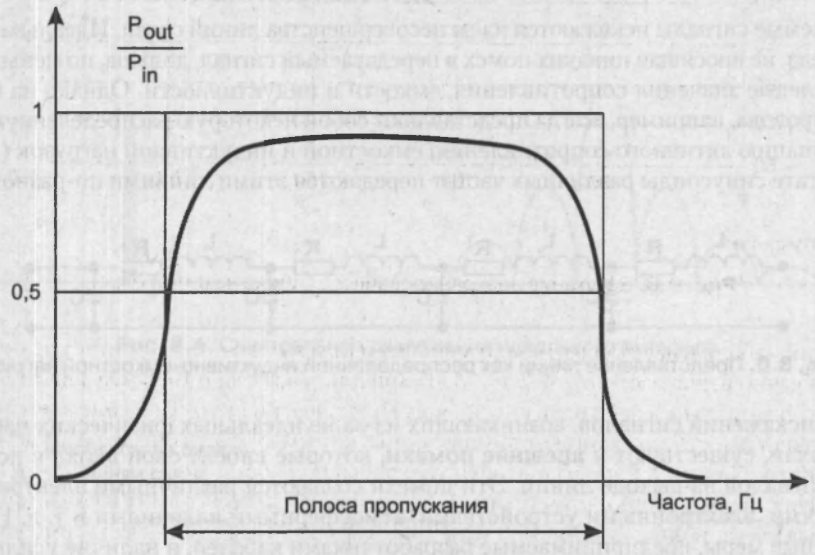


Рис. 8.7. Зависимость затухания от частоты

ВНИМАНИЕ

Как было сказано, затухание всегда имеет отрицательное значение, однако знак минус часто опускают, при этом иногда возникает путаница. Совершенно корректно утверждение, что качество линии связи тем выше, чем больше (с учетом знака) затухание. Если же игнорировать знак, то есть иметь в виду абсолютное значение затухания, то у более качественной линии затухание меньше. Приведем пример. Для внутренней проводки в зданиях используется кабель на витой паре категории 5. Этот кабель, на котором работают практически все технологии локальных сетей, характеризуется затуханием не меньше, чем –23,6 дБ для частоты 100 МГц при длине кабеля 100 м. Более качественный кабель категории 6 имеет на частоте 100 МГц затухание не меньше, чем –20,6 дБ. Получаем, что – 20,6 > –23,6, но 20,6 < 23,6.

Чаще всего при описании параметров линии связи приводятся значения затухания всего *для нескольких значений частот*. Это объясняется, с одной стороны, стремлением упростить измерения при проверке качества линии. С другой стороны, на практике часто заранее известна основная частота передаваемого сигнала, то есть та частота, гармоника которой имеет наибольшую амплитуду и мощность. Поэтому достаточно знать затухание на этой частоте, чтобы приблизительно оценить искажения передаваемых по линии сигналов.

На рис. 8.8 показаны типовые зависимости затухания от частоты для кабелей на неэкранированной витой паре категорий 5 и 6.

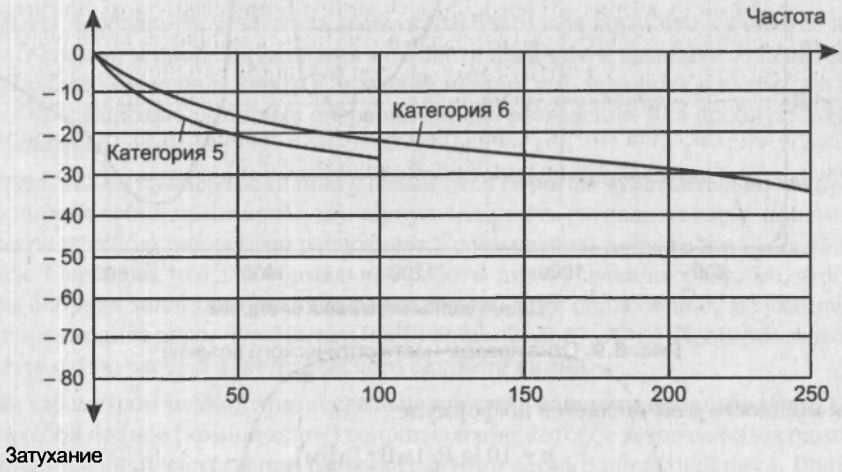


Рис. 8.8. Затухание неэкранированного кабеля на витой паре

Оптический кабель имеет существенно меньшие (по абсолютной величине) величины затухания, обычно в диапазоне от –0,2 до –3 дБ при длине кабеля в 1000 м, а значит, является более качественным, чем кабель на витой паре. Практически все оптические волокна имеют сложную зависимость затухания от длины волны, которая имеет три так называемых **окна прозрачности**. На рис. 8.9 показана характерная зависимость затухания для оптического волокна. Из рисунка видно, что область эффективного использования современных волокон ограничена волнами длин 850 нм, 1300 нм и 1550 нм (соответственно частотами 35 ТГц, 23 ТГц и 19,4 ТГц). Окно 1550 нм обеспечивает наименьшие потери, а значит, максимальную дальность при фиксированной мощности передатчика и фиксированной чувствительности приемника.

В качестве характеристики мощности сигнала используются абсолютный и относительный уровни мощности. **Абсолютный уровень мощности** измеряется в ваттах, **относительный уровень мощности**, как и затухание, измеряется в децибелах.

Существует также и другая абсолютная единица измерения мощности — так называемая **опорная мощность**, измеряемая в децибелах на милливатт (дБм).

При определении опорной мощности также используется логарифм отношения мощностей, но значение мощности, к которой выполняется отношение, *фиксируется*. Опорный уровень мощности, к которой относится измеряемая мощность, принимается равным 1 мВт, что и отражается в названии этой единицы мощности.

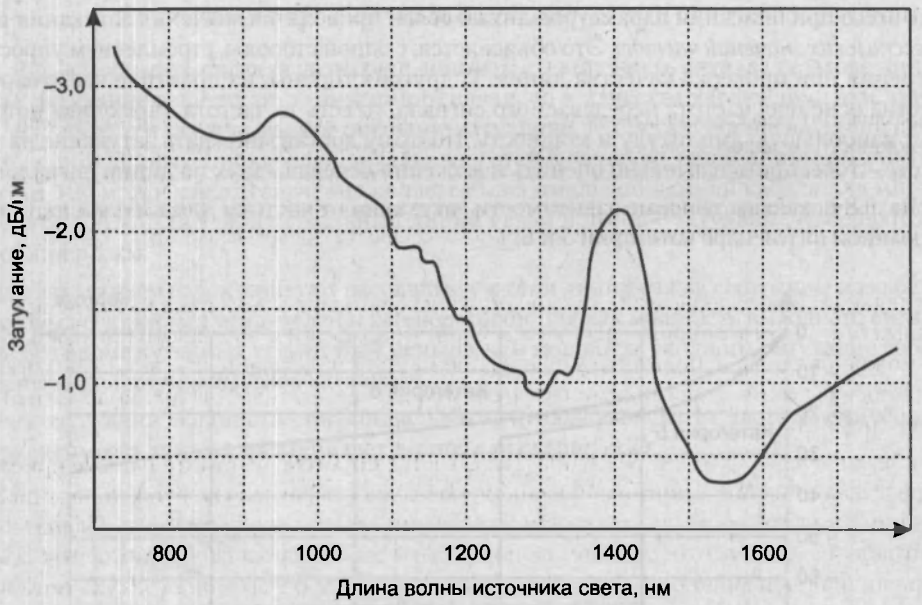


Рис. 8.9. Окна прозрачности оптического волокна

Опорная мощность p вычисляется по формуле:

$$p = 10 \lg P/1\text{мВт}\ [\text{дБм}].$$

Здесь P — абсолютная мощность сигнала в милливаттах.

Несмотря на использование отношения в определении опорной мощности, эта единица измерения является *абсолютной*, а не относительной, так как однозначно преобразует абсолютную мощность сигнала в ваттах в некоторое значение, которое никак не зависит от значения мощности другого сигнала, как это имеет место при определении децибела. Так, нетрудно вычислить соответствие некоторых значений мощности сигнала, выраженные в ваттах и децибелах на милливатт:

1 мВ = 0 дБм;

10 мВ = 10 дБм;

1 В = 30 дБм;

100 кВ = 80 дБм.

Опорные значения мощности удобно использовать при *расчетах энергетического бюджета линий связи*.

ПРИМЕР

Пусть требуется определить минимальную опорную мощность x (дБм) передатчика, достаточную для того, чтобы на выходе линии опорная мощность сигнала была не ниже некоторого порогового значения y (дБм). Затухание линии известно и равно A. Пусть X и Y — это абсолютные значения мощности сигнала, заданные в милливаттах на входе и выходе линии соответственно.

По определению $A = 10 \lg X/Y$. Используя свойства логарифмов, имеем:

$$A = 10 \lg X/Y = 10\lg(X/1)/(Y/1) = 10 \lg X/1 \text{ мВт} - 10 \lg Y/1 \text{ мВт}.$$

Заметим, что два последних члена уравнения по определению являются опорными значениями мощности сигналов на выходе и входе, поэтому приходим к простому соотношению $A = x - y$, где x — опорная мощность входного сигнала, а y — опорная мощность выходного сигнала.

Из последнего соотношения следует, что минимальная требуемая мощность передатчика может быть определена как сумма затухания и мощности сигнала на выходе: $x = A + y$.

Предельная простота расчета стала возможной благодаря тому, что в качестве исходных данных были взяты опорные значения мощности входного и выходного сигналов. Конечно, можно было бы использовать и значение мощностей, заданных в ваттах, но при этом пришлось бы заниматься такими операциями, как возведение 10 в дробную степень, что более громоздко.

Использованная в примере величина y называется **порогом чувствительности приемника** и представляет собой минимальную опорную мощность сигнала на входе приемника, при котором он способен корректно распознавать дискретную информацию, содержащуюся в сигнале. Очевидно, что для нормальной работы линии связи необходимо, чтобы минимальная опорная мощность сигнала передатчика, даже ослабленная затуханием линии связи, превосходила порог чувствительности приемника: $x - A > y$. Проверка этого условия и является сутью расчета энергетического бюджета линии.

Важным параметром медной линии связи является ее **волновое сопротивление**, представляющее собой полное (комплексное) сопротивление, которое встречает электромагнитная волна определенной частоты при распространении вдоль однородной цепи. Волновое сопротивление измеряется в омах и зависит от таких параметров линии связи, как активное сопротивление, погонная индуктивность и погонная емкость, а также от частоты самого сигнала. Выходное сопротивление передатчика должно быть согласовано с волновым сопротивлением линии, иначе затухание сигнала будет чрезмерно большим.

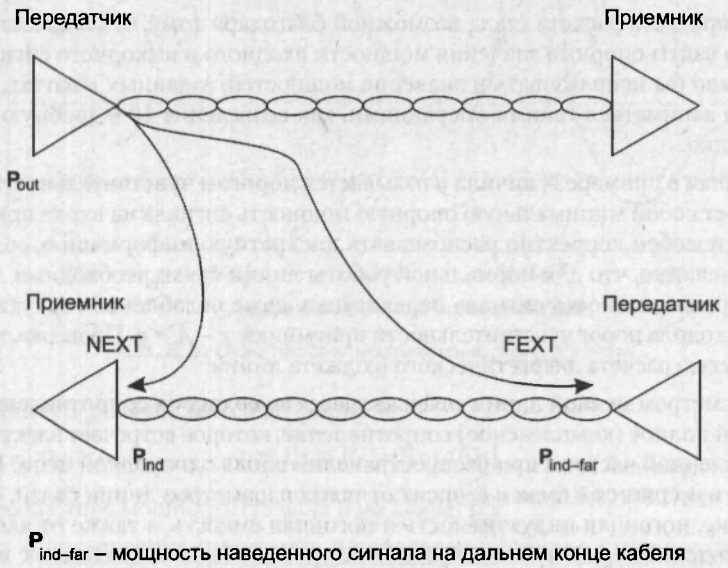
Помехоустойчивость и достоверность

Помехоустойчивость линии, как и следует из названия, определяет способность линии противостоять влиянию помех, создаваемых во внешней среде или на внутренних проводниках самого кабеля. Помехоустойчивость линии зависит от типа используемой физической среды, а также от средств экранирования и подавления помех самой линии. Наименее помехоустойчивыми являются радиолинии, хорошей устойчивостью обладают кабельные линии и отличной — волоконно-оптические линии, мало чувствительные к внешнему электромагнитному излучению. Обычно для уменьшения помех, создаваемых внешними электромагнитными полями, проводники экранируют и/или скручивают.

Электрическая и магнитная связь — это параметры медного кабеля, также являющиеся результатом помех. **Электрическая связь** определяется отношением наведенного тока в подверженной влиянию цепи к напряжению, действующему во влияющей цепи. **Магнитная связь** — это отношение электродвижущей силы, наведенной в подверженной влиянию цепи, к току во влияющей цепи. Результатом электрической и магнитной связи являются

наведенные сигналы (наводки) в цепи, подверженной вилянию. Существует несколько различных параметров, характеризующих устойчивость кабеля к наводкам.

Перекрестные наводки на ближнем конце (Near End Cross Talk, NEXT) определяют устойчивость кабеля в том случае, когда наводка образуется в результате действия сигнала, генерируемого передатчиком, подключенным к одной из соседних пар на том же конце кабеля, на котором работает подключенный к подверженной влиянию паре приемник (рис. 8.10). Показатель NEXT, выраженный в децибелах, равен 10 lg P_{out}/F_{ind}, где P_{out} — мощность выходного сигнала, P_{ind} — мощность наведенного сигнала.



$P_{ind-far}$ – мощность наведенного сигнала на дальнем конце кабеля

Рис. 8.10. Переходное затухание

Чем меньше значение NEXT, тем лучше кабель. Так, для витой пары категории 5 показатель NEXT должен быть меньше –27 дБ на частоте 100 МГц.

Перекрестные наводки на дальнем конце (Far End Cross Talk, FEXT) позволяют оценить устойчивость кабеля к наводкам для случая, когда передатчик и приемник подключены к разным концам кабеля. Очевидно, что этот показатель должен быть лучше, чем NEXT, так как до дальнего конца кабеля сигнал приходит ослабленный затуханием каждой пары.

Показатели NEXT и FEXT обычно применяются к кабелю, состоящему из нескольких витых пар, так как в этом случае взаимные наводки одной пары на другую могут достигать значительных величин. Для одинарного коаксиального кабеля (то есть состоящего из одной экранированной жилы) этот показатель не имеет смысла, а для двойного коаксиального кабеля он также не применяется вследствие высокой степени защищенности каждой жилы. Оптические волокна тоже не создают сколько-нибудь заметных взаимных помех.

В связи с тем, что в некоторых новых технологиях данные передаются одновременно по нескольким витым парам, в последнее время стали применяться также показатели перекрестных наводок с приставкой PS (PowerSUM — объединенная наводка), такие как PS NEXT и PS FEXT. Эти показатели отражают устойчивость кабеля к суммарной мощ-

ности перекрестных наводок на одну из пар кабеля от всех остальных передающих пар (рис. 8.11).

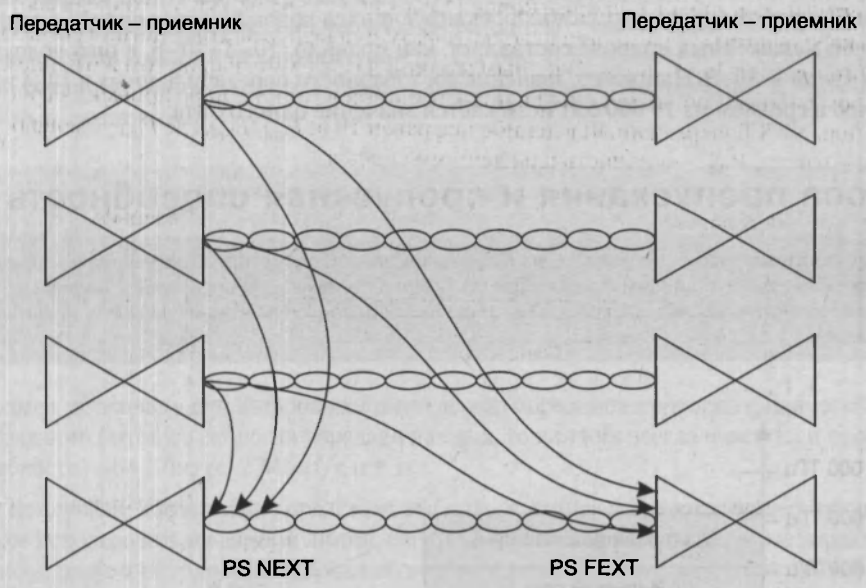


Рис. 8.11. Суммарное переходное затухание

Еще одним практически важным показателем является **защищенность кабеля** (Attenuation/Crosstalk Ratio, ACR). Защищенность определяется как разность между уровнями полезного сигнала и помех. Чем больше значение защищенности кабеля, тем в соответствии с формулой Шеннона данные можно передавать по этому кабелю с потенциально более высокой скоростью. На рис. 8.12 показана типичная характеристика зависимости защищенности кабеля на неэкранированной витой паре от частоты сигнала.

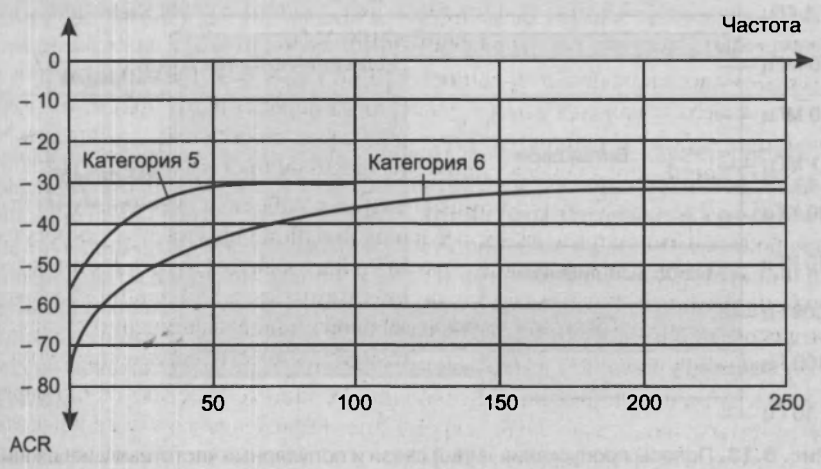


Рис. 8.12. Защищенность витой пары

Достоверность передачи данных характеризует вероятность искажения каждого передаваемого бита данных. Иногда этот же показатель называют **интенсивностью битовых ошибок** (Bit Error Rate, BER). Величина BER для линий связи без дополнительных средств защиты от ошибок (например, самокорректирующихся кодов или протоколов с повторной передачей искаженных кадров) составляет, как правило, 10–4–10–6, в оптоволоконных линиях связи — 10–9. Например, значение достоверности передачи данных в 10–4 говорит о том, что в среднем из 10 000 бит искажается значение одного бита.

Полоса пропускания и пропускная способность

Полоса пропускания — это непрерывный диапазон частот, для которого затухание не превышает некоторый заранее заданный предел. То есть полоса пропускания определяет диапазон частот синусоидального сигнала, при которых этот сигнал передается по линии связи без значительных искажений.

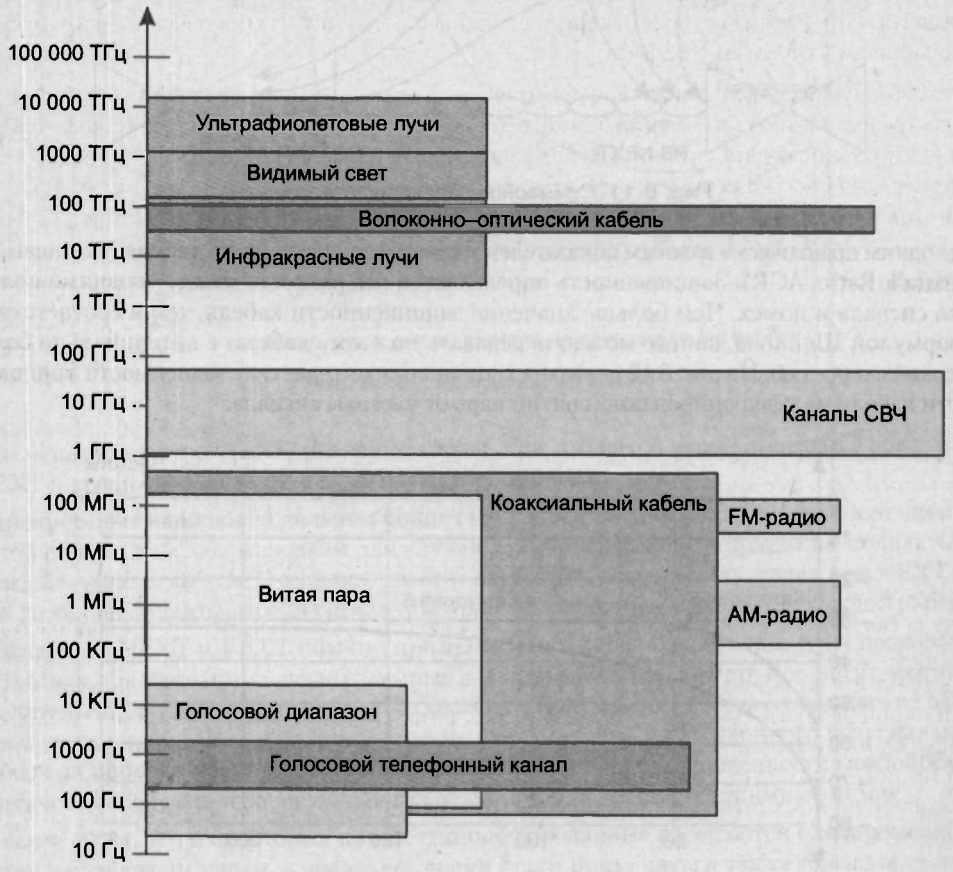


Рис. 8.13. Полосы пропускания линий связи и популярные частотные диапазоны

Часто граничными частотами считаются частоты, на которых мощность выходного сигнала уменьшается в два раза по отношению к входному, что соответствует затуханию в –3 дБ. Как мы увидим далее, *ширина* полосы пропускания в наибольшей степени влияет на максимально возможную скорость передачи информации по линии связи. Полоса пропускания зависит от типа линии и ее протяженности. На рис. 8.13 показаны полосы пропускания линий связи различных типов, а также наиболее часто используемые в технике связи частотные диапазоны.

Пропускная способность линии характеризует максимально возможную скорость передачи данных, которая может быть достигнута на этой линии. Особенностью пропускной способности является то, что, с одной стороны, эта характеристика зависит от параметров *физической среды*, а с другой — определяется *способом передачи данных*. Следовательно, нельзя говорить о пропускной способности линии связи до того, как для нее определен протокол физического уровня.

Например, поскольку для цифровых линий всегда определен протокол физического уровня, задающий битовую скорость передачи данных, то для них всегда известна и пропускная способность — 64 Кбит/с, 2 Мбит/с и т. п.

В тех же случаях, когда только предстоит выбрать, какой из множества существующих протоколов использовать на данной линии, очень важными являются остальные характеристики линии, такие как полоса пропускания, перекрестные наводки, помехоустойчивость и др. Пропускная способность, как и скорость передачи данных, измеряется в битах в секунду (бит/с), а также в производных единицах, таких как килобиты в секунду (Кбит/с) и т. д.

ВНИМАНИЕ

Пропускная способность линий связи и коммуникационного сетевого оборудования традиционно измеряется в битах в секунду, а не в байтах в секунду. Это связано с тем, что данные в сетях передаются последовательно, то есть побитно, а не параллельно, байтами, как это происходит между устройствами внутри компьютера. Такие единицы измерения, как килобит, мегабит или гигабит, в сетевых технологиях строго соответствуют степеням десяти (то есть килобит — это 1000 бит, а мегабит — это 1 000 000 бит), как это принято во всех отраслях науки и техники, а не близким к этим числам степеням двойки, как это принято в программировании, где приставка «кило» равна $2^{10} = 1024$, а «мега» — $2^{20} = 1\,048\,576$.

Пропускная способность линии связи зависит не только от ее характеристик, таких как затухание и полоса пропускания, но и от спектра передаваемых сигналов. Если значимые гармоники сигнала (то есть те гармоники, амплитуды которых вносят основной вклад в результирующий сигнал) попадают в полосу пропускания линии, то такой сигнал будет хорошо передаваться данной линией связи, и приемник сможет правильно распознать информацию, отправленную по линии передатчиком (рис. 8.14, *а*). Если же значимые гармоники выходят за границы полосы пропускания линии связи, то сигнал начнет значительно искажаться, и приемник будет ошибаться при распознавании информации (рис. 8.14, *б*).

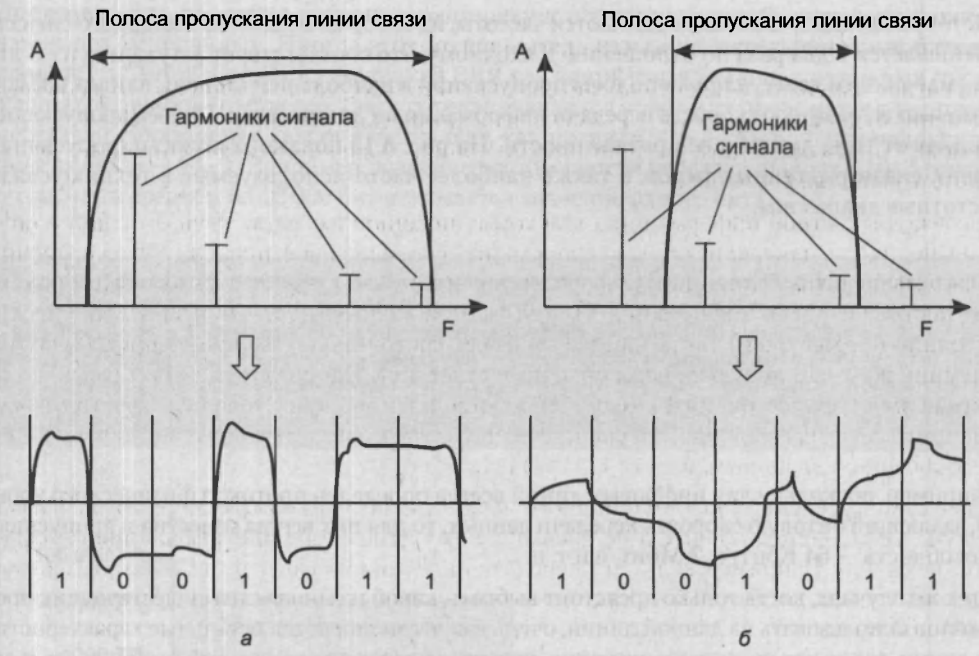


Рис. 8.14. Соответствие между полосой пропускания линии связи и спектром сигнала

Биты и боды

Выбор способа представления дискретной информации в виде сигналов, подаваемых на линию связи, называется **физическим**, или **линейным**, **кодированием**. От выбранного способа кодирования зависит спектр сигналов и, соответственно, пропускная способность линии.

Таким образом, для одного способа кодирования линия может обладать одной пропускной способностью, а для другого — другой. Например, витая пара категории 3 может передавать данные с пропускной способностью 10 Мбит/с при способе кодирования стандарта физического уровня 10Base-T и 33 Мбит/с при способе кодирования стандарта 100Base-T4.

ВНИМАНИЕ

В соответствии с основным постулатом теории информации любое различимое непредсказуемое изменение принимаемого сигнала несет в себе информацию. Отсюда следует, что синусоида, у которой амплитуда, фаза и частота остаются неизменными, информации не несет, так как изменение сигнала хотя и происходит, но является абсолютно предсказуемым. Аналогично, не несут в себе информации импульсы на тактовой шине компьютера, так как их изменения тоже постоянны во времени. А вот импульсы на шине данных предсказать заранее нельзя, это и делает их информационными, они переносят информацию между отдельными блоками или устройствами компьютера.

В большинстве способов кодирования используется изменение какого-либо параметра периодического сигнала — частоты, амплитуды и фазы синусоиды или же знака потенциала последовательности импульсов. Периодический сигнал, параметры которого подвергаются изменениям, называют **несущим сигналом**, а его частоту, если сигнал синусоидальный, —

несущей частотой. Процесс изменения параметров несущего сигнала в соответствии с передаваемой информацией называется **модуляцией**.

Если сигнал изменяется так, что можно различить только два его состояния, то любое его изменение будет соответствовать наименьшей единице информации — биту. Если же сигнал может иметь более двух различимых состояний, то любое его изменение будет нести *несколько битов информации*.

Передача дискретной информации в телекоммуникационных сетях осуществляется тактировано, то есть изменение сигнала происходит через фиксированный интервал времени, называемый **тактом**. Приемник информации считает, что в начале каждого такта на его вход поступает новая информация. При этом независимо от того, повторяет ли сигнал состояние предыдущего такта или же он имеет состояние, отличное от предыдущего, приемник получает новую информацию от передатчика. Например, если такт равен 0,3 с, а сигнал имеет два состояния и 1 кодируется потенциалом 5 вольт, то присутствие на входе приемника сигнала величиной 5 вольт в течение 3 секунд означает получение информации, представленной двоичным числом 1111111111.

Количество изменений информационного параметра несущего периодического сигнала в секунду измеряется в **бодах**. 1 бод равен одному изменению информационного параметра в секунду. Например, если такт передачи информации равен 0,1 секунды, то сигнал изменяется со скоростью 10 бод. Таким образом, скорость в бодах целиком определяется величиной такта.

Информационная скорость измеряется в битах в секунду и в общем случае *не совпадает* со скоростью в бодах. Она может быть как выше, так и ниже скорости изменения информационного параметра, измеряемого в *бодах*. Это соотношение зависит от числа состояний сигнала. Например, если сигнал имеет более двух различимых состояний, то при равных тактах и соответствующем методе кодирования информационная скорость в битах в секунду может быть *выше*, чем скорость изменения информационного сигнала в бодах.

Пусть информационными параметрами являются фаза и амплитуда синусоиды, причем различаются 4 состояния фазы в 0, 90, 180 и 270° и два значения амплитуды сигнала — тогда информационный сигнал может иметь 8 различимых состояний. Это означает, что любое состояние этого сигнала несет информацию в 3 бит. В этом случае модем, работающий со скоростью 2400 бод (меняющий информационный сигнал 2400 раз в секунду), передает информацию со скоростью 7200 бит/с, так как при одном изменении сигнала передается 3 бита информации.

Если сигнал имеет два состояния (то есть несет информацию в 1 бит), то информационная скорость обычно совпадает с количеством бодов. Однако может наблюдаться и обратная картина, когда информационная скорость оказывается *ниже* скорости изменения информационного сигнала в бодах. Это происходит в тех случаях, когда для надежного распознавания приемником пользовательской информации каждый бит в последовательности кодируется несколькими изменениями информационного параметра несущего сигнала. Например, при кодировании единичного значения бита импульсом положительной полярности, а нулевого значения бита импульсом отрицательной полярности физический сигнал дважды изменяет свое состояние при передаче каждого бита. При таком кодировании скорость линии в битах в секунду в два раза ниже, чем в бодах.

Чем выше частота несущего периодического сигнала, тем выше может быть частота модуляции и тем выше может быть пропускная способность линии связи.

Однако с увеличением частоты периодического несущего сигнала увеличивается и ширина спектра этого сигнала.

Линия передает этот спектр синусоид с теми искажениями, которые определяются ее полосой пропускания. Чем больше несоответствие между полосой пропускания линии и шириной спектра передаваемых информационных сигналов, тем больше сигналы искажаются и тем вероятнее ошибки в распознавании информации принимающей стороной, а значит, возможная скорость передачи информации оказывается меньше.

Соотношение полосы пропускания и пропускной способности

Связь между полосой пропускания линии и ее пропускной способностью вне зависимости от принятого способа физического кодирования установил *Клод Шеннон*:

$$C = F \log_2 (1 + P_с/P_ш).$$

Здесь C — пропускная способность линии в битах в секунду, F — ширина полосы пропускания линии в герцах, $P_с$ — мощность сигнала, $P_ш$ — мощность шума.

Из этого соотношения следует, что теоретического предела пропускной способности линии с фиксированной полосой пропускания не существует. Однако на практике такой предел имеется. Действительно, повысить пропускную способность линии можно за счет увеличения мощности передатчика или же уменьшения мощности шума (помех) в линии связи. Обе эти составляющие поддаются изменению с большим трудом. Повышение мощности передатчика ведет к значительному увеличению его габаритов и стоимости. Снижение уровня шума требует применения специальных кабелей с хорошими защитными экранами, что весьма дорого, а также снижения шума в передатчике и промежуточной аппаратуре, чего достичь весьма не просто. К тому же влияние мощностей полезного сигнала и шума на пропускную способность ограничено логарифмической зависимостью, которая растет далеко не так быстро, как прямо-пропорциональная. Так, при достаточно типичном исходном отношении мощности сигнала к мощности шума в 100 раз повышение мощности передатчика в 2 раза даст только 15 % увеличения пропускной способности линии.

Близким по сути к формуле Шеннона является другое соотношение, полученное *Найквистом*, которое также определяет максимально возможную пропускную способность линии связи, но без учета шума в линии:

$$C = 2F \log_2 M.$$

Здесь M — количество различимых состояний информационного параметра.

Если сигнал имеет два различимых состояния, то пропускная способность равна удвоенному значению ширины полосы пропускания линии связи (рис. 8.15, *а*). Если же в передатчике используется более двух устойчивых состояний сигнала для кодирования данных, то пропускная способность линии повышается, так как за один такт работы передатчик передает несколько битов исходных данных, например 2 бита при наличии четырех различимых состояний сигнала (рис. 8.15, *б*).

Хотя в формуле Найквиста наличие шума в явном виде не учитывается, косвенно его влияние отражается в выборе количества состояний информационного сигнала. Для повышения

пропускной способности линии связи следовало бы увеличивать количество состояний, но на практике этому препятствует шум на линии. Например, пропускную способность линии, сигнал которой показан на рис. 8.15, *б*, можно увеличить еще в два раза, применив для кодирования данных не 4, а 16 уровней. Однако если амплитуда шума время от времени превышает разницу между соседними уровнями, то приемник не сможет устойчиво распознавать передаваемые данные. Поэтому количество возможных состояний сигнала фактически ограничивается соотношением мощности сигнала и шума, а формула Найквиста определяет предельную скорость передачи данных в том случае, когда количество состояний уже выбрано с учетом возможностей устойчивого распознавания приемником.

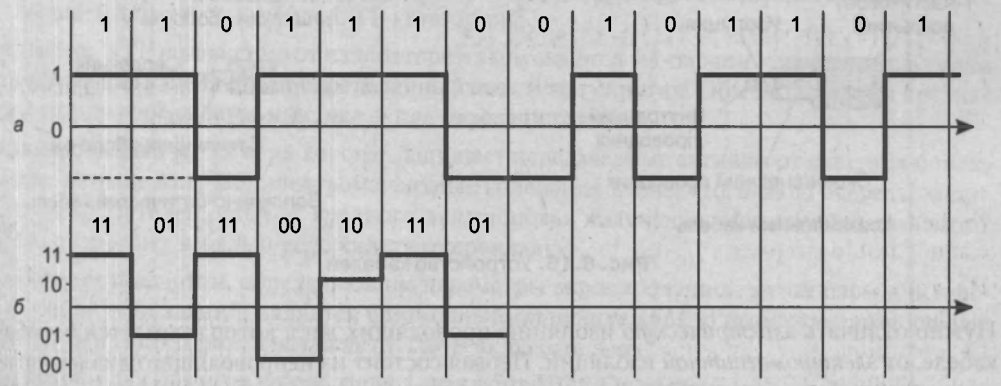


Рис. 8.15. Повышение скорости передачи за счет дополнительных состояний сигнала

Типы кабелей

Сегодня как для внутренней (кабели зданий), так и для внешней проводки чаще всего применяются три класса проводных линий связи:

- витая пара;
- коаксиальные кабели;
- волоконно-оптические кабели.

Экранированная и неэкранированная витая пара

Витой парой называется скрученная пара проводов. Этот вид среды передачи данных очень популярен и составляет основу большого количества как внутренних, так и внешних кабелей. Кабель может состоять из нескольких скрученных пар (внешние кабели иногда содержат до нескольких десятков таких пар).

Скручивание проводов снижает влияние внешних и взаимных помех на полезные сигналы, передаваемые по кабелю.

Основные особенности конструкции кабелей схематично показаны на рис. 8.16.

Кабели на основе витой пары являются *симметричными*, то есть они состоят из двух одинаковых в конструктивном отношении проводников. Симметричный кабель на основе витой пары может быть как *экранированным*, так и *неэкранированным*.

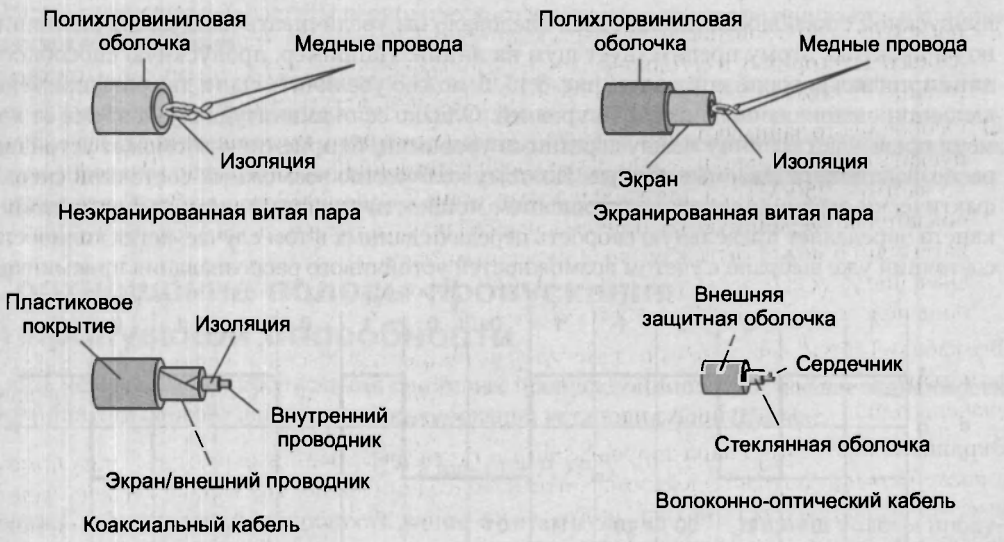


Рис. 8.16. Устройство кабелей

Нужно отличать *электрическую* изоляцию проводящих жил, которая имеется в любом кабеле, от *электромагнитной* изоляции. Первая состоит из непроводящего диэлектрического слоя — бумаги или полимера, например поливинилхлорида или полистирола. Во втором случае помимо электрической изоляции проводящие жилы помещаются также внутрь электромагнитного экрана, в качестве которого чаще всего применяется проводящая медная оплетка.

Кабель на основе **неэкранированной витой пары**, используемый для проводки внутри здания, разделяется в международных стандартах на *категории* (от 1 до 7).

- Кабели *категории 1* применяются там, где требования к скорости передачи минимальны. Обычно это кабель для цифровой и аналоговой передачи голоса и низкоскоростной (до 20 Кбит/с) передачи данных. До 1983 года это был основной тип кабеля для телефонной разводки.
- Кабели *категории 2* были впервые применены фирмой IBM при построении собственной кабельной системы. Главное требование к кабелям этой категории — способность передавать сигналы со спектром до 1 МГц.
- Кабели *категории 3* были стандартизованы в 1991 году. Стандарт EIA-568 определил электрические характеристики кабелей для частот в диапазоне до 16 МГц. Кабели категории 3, предназначенные как для передачи данных, так и для передачи голоса, составляют сейчас основу многих кабельных систем зданий.
- Кабели *категории 4* представляют собой несколько улучшенный вариант кабелей категории 3. Кабели категории 4 обязаны выдерживать тесты на частоте передачи сигнала 20 МГц и обеспечивать повышенную помехоустойчивость и низкие потери сигнала. На практике используются редко.
- Кабели *категории 5* были специально разработаны для поддержки высокоскоростных протоколов. Их характеристики определяются в диапазоне до 100 МГц. Большинство высокоскоростных технологий (FDDI, Fast Ethernet, ATM и Gigabit Ethernet) ориенти-

ровано на использование витой пары категории 5. Кабель категории 5 пришел на замену кабелю категории 3, и сегодня все новые кабельные системы крупных зданий строятся именно на этом типе кабеля (в сочетании с волоконно-оптическим).

- Особое место занимают кабели *категорий 6 и 7*, которые промышленность начала выпускать сравнительно недавно. Для кабеля категории 6 характеристики определяются до частоты 250 МГц, а для кабелей категории 7 — до 600 МГц. Кабели категории 7 обязательно экранируются, причем как каждая пара, так и весь кабель в целом. Кабель категории 6 может быть как экранированным, так и неэкранированным. Основное назначение этих кабелей — поддержка высокоскоростных протоколов на отрезках кабеля большей длины, чем кабель UTP категории 5.

Все кабели UTP независимо от их категории выпускаются в 4-парном исполнении. Каждая из четырех пар кабеля имеет определенный цвет и шаг скрутки. Обычно две пары предназначены для передачи данных, две — для передачи голоса.

Экранированная витая пара хорошо защищает передаваемые сигналы от внешних помех, а также меньше излучает электромагнитные колебания вовне, что, в свою очередь, защищает пользователей сетей от вредного для здоровья излучения. Наличие заземляемого экрана удорожает кабель и усложняет его прокладку.

Основным стандартом, определяющим параметры экранированной витой пары для применения внутри зданий, является фирменный *стандарт IBM*. В этом стандарте кабели делятся не на категории, а на *типы* от 1 до 9 включительно.

Рассмотрим для примера кабель *типа 1* стандарта IBM. Он состоит из 2-х пар скрученных проводов, экранированных проводящей оплеткой, которая заземляется. Электрические параметры кабеля типа 1 примерно соответствуют параметрам кабеля UTP категории 5. Однако волновое сопротивление кабеля типа 1, равное 150 Ом, значительно выше волнового сопротивления UTP категории 5 (100 Ом), поэтому невозможно «улучшение» кабельной проводки сети путем простой замены неэкранированной пары экранированной парой типа 1. Передатчики, рассчитанные на работу с кабелем, имеющим волновое сопротивление 100 Ом, будут плохо работать на волновое сопротивление 150 Ом.

Коаксиальный кабель

Коаксиальный кабель состоит из несимметричных пар проводников. Каждая пара представляет собой внутреннюю медную жилу и соосную с ней внешнюю жилу, которая может быть полой медной трубой или оплеткой, отделенной от внутренней жилы диэлектрической изоляцией. Внешняя жила играет двоякую роль — по ней передаются информационные сигналы и она является экраном, защищающим внутреннюю жилу от внешних электромагнитных полей. Существует несколько типов коаксиального кабеля, отличающихся характеристиками и областями применения: для локальных компьютерных сетей, для глобальных телекоммуникационных сетей, для кабельного телевидения и т. п.

Согласно современным стандартам коаксиальный кабель не считается хорошим выбором при построении структурированной кабельной системы зданий. Далее приводятся основные типы и характеристики этих кабелей.

- **«Толстый» коаксиальный кабель** разработан для сетей Ethernet 10Base-5 с волновым сопротивлением 50 Ом и внешним диаметром около 12 мм. Этот кабель имеет достаточно толстый внутренний проводник диаметром 2,17 мм, который обеспечивает хорошие

механические и электрические характеристики (затухание на частоте 10 МГц — не хуже 18 дБ/км). Зато этот кабель сложно монтировать — он плохо гнется.

- **«Тонкий» коаксиальный кабель** предназначен для сетей Ethernet 10Base-2. Обладая внешним диаметром около 50 мм и тонким внутренним проводником 0,89 мм, этот кабель не так прочен, как «толстый» коаксиал, зато обладает гораздо большей гибкостью, что удобно при монтаже. «Тонкий» коаксиальный кабель также имеет волновое сопротивление 50 Ом, но его механические и электрические характеристики хуже, чем у «толстого» коаксиального кабеля. Затухание в этом типе кабеля выше, чем в «толстом» коаксиальном кабеле, что приводит к необходимости уменьшать длину кабеля для получения одинакового затухания в сегменте.
- **Телевизионный кабель** с волновым сопротивлением 75 Ом широко применяется в кабельном телевидении. Существуют стандарты локальных сетей, позволяющие использовать такой кабель для передачи данных.

Волоконно-оптический кабель

Волоконно-оптический кабель состоит из тонких (5–60 микрон) гибких стеклянных волокон (волоконных световодов), по которым распространяются световые сигналы. Это наиболее качественный тип кабеля — он обеспечивает передачу данных с очень высокой скоростью (до 10 Гбит/с и выше) и к тому же лучше других типов передающей среды обеспечивает защиту данных от внешних помех (в силу особенностей распространения света такие сигналы легко экранировать).

Каждый световод состоит из центрального проводника света (сердцевины) — стеклянного волокна, и стеклянной оболочки, обладающей меньшим показателем преломления, чем сердцевина. Распространяясь по сердцевине, лучи света не выходят за ее пределы, отражаясь от покрывающего слоя оболочки. В зависимости от распределения показателя преломления и величины диаметра сердечника различают:

- многомодовое волокно со ступенчатым изменением показателя преломления (рис. 8.17, *а*);
- многомодовое волокно с плавным изменением показателя преломления (рис. 8.17, *б*);
- одномодовое волокно (рис. 8.17, *в*).

Понятие «мода» описывает режим распространения световых лучей в сердцевине кабеля.

В **одномодовом кабеле** (Single Mode Fiber, SMF) используется центральный проводник очень малого диаметра, соизмеримого с длиной волны света — от 5 до 10 мкм. При этом практически все лучи света распространяются вдоль оптической оси световода, не отражаясь от внешнего проводника. Изготовление сверхтонких качественных волокон для одномодового кабеля представляет собой сложный технологический процесс, что делает одномодовый кабель достаточно дорогим. Кроме того, в волокно такого маленького диаметра достаточно сложно направить пучок света, не потеряв при этом значительную часть его энергии.

В **многомодовых кабелях** (Multi Mode Fiber, MMF) используются более широкие внутренние сердечники, которые легче изготовить технологически. В многомодовых кабелях во внутреннем проводнике одновременно существует несколько световых лучей, отражающихся от внешнего проводника под разными углами. Угол отражения луча называется **модой** луча. В многомодовых кабелях с плавным изменением коэффициента преломления

режим отражения лучей имеет сложный характер. Возникающая при этом интерференция ухудшает качество передаваемого сигнала, что приводит к искажениям передаваемых импульсов в многомодовом оптическом волокне. По этой причине технические характеристики многомодовых кабелей хуже, чем одномодовых.

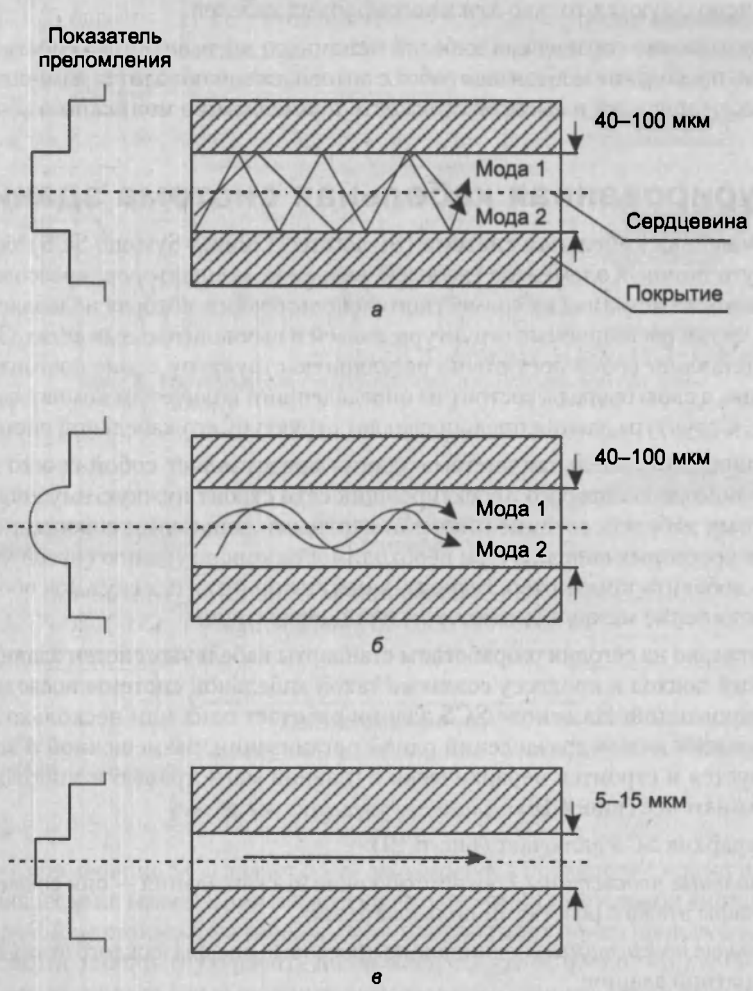


Рис. 8.17. Типы оптического кабеля

Учитывая это, многомодовые кабели применяют в основном для передачи данных на скоростях не более 1 Гбит/с на небольшие расстояния (до 300–2000 м), а одномодовые — для передачи данных со сверхвысокими скоростями в несколько десятков гигабитов в секунду (а при использовании технологии DWDM — до нескольких терабитов в секунду) на расстояния до нескольких десятков и даже сотен километров (дальняя связь).

В качестве источников света в волоконно-оптических кабелях применяются:

- светодиоды, или светоизлучающие диоды (Light Emitted Diode, LED);
- полупроводниковые лазеры, или лазерные диоды.

Для одномодовых кабелей применяются только лазерные диоды, так как при таком малом диаметре оптического волокна световой поток, создаваемый светодиодом, невозможно без больших потерь направить в волокно — он имеет чересчур широкую диаграмму направленности излучения, в то время как лазерный диод — узкую. Более дешевые светодиодные излучатели используются только для многомодовых кабелей.

Стоимость волоконно-оптических кабелей ненамного превышает стоимость кабелей на витой паре, но проведение монтажных работ с оптоволокном обходится намного дороже из-за трудоемкости операций и высокой стоимости применяемого монтажного оборудования.

Структурированная кабельная система зданий

Структурированная кабельная система (Structured Cabling System, SCS) здания — это набор коммутационных элементов (кабелей, разъемов, коннекторов, кроссовых панелей и шкафов), а также методика их совместного использования, которая позволяет создавать регулярные, легко расширяемые структуры связей в вычислительных сетях. Здание само по себе представляет собой достаточно регулярную структуру — оно состоит из этажей, а каждый этаж, в свою очередь, состоит из определенного количества комнат, соединенных коридорами. Структура здания предопределяет структуру его кабельной системы.

Структурированная кабельная система здания представляет собой своего рода «конструктор», с помощью которого проектировщик сети строит нужную ему конфигурацию из стандартных кабелей, соединенных стандартными разъемами и коммутируемых на стандартных кроссовых панелях. При необходимости конфигурацию связей можно легко изменить — добавить компьютер, сегмент, коммутатор, изъять ненужное оборудование, поменять соединение между компьютером и концентратором.

Наиболее детально на сегодня разработаны стандарты кабельных систем зданий, при этом иерархический подход к процессу создания такой кабельной системы позволяет назвать ее структурированной. На основе SCS здания работает одна или несколько локальных сетей организаций или подразделений одной организации, размещенной в этом здании. SCS планируется и строится иерархически с главной магистралью и многочисленными ответвлениями от нее (рис. 8.18).

Типичная иерархия SCS включает (рис. 8.19):

- *горизонтальные подсистемы*, соответствующие этажам здания — они соединяют кроссовые шкафы этажа с розетками пользователей;
- *вертикальные подсистемы*, соединяющие кроссовые шкафы каждого этажа с центральной аппаратной здания;
- *подсистема кампуса*, объединяющая несколько зданий с главной аппаратной всего кампуса (эта часть кабельной системы обычно называется магистралью).

Использование структурированной кабельной системы вместо хаотически проложенных кабелей дает предприятию много преимуществ. Система SCS при продуманной организации может стать *универсальной средой* для передачи компьютерных данных в локальной вычислительной сети, организации локальной телефонной сети, передачи видеоинформации и даже для передачи сигналов от датчиков пожарной безопасности или охранных систем. Подобная универсализация позволяет автоматизировать многие процессы контроля, мониторинга и управления хозяйственными службами и системами жизнеобеспечения предприятия.

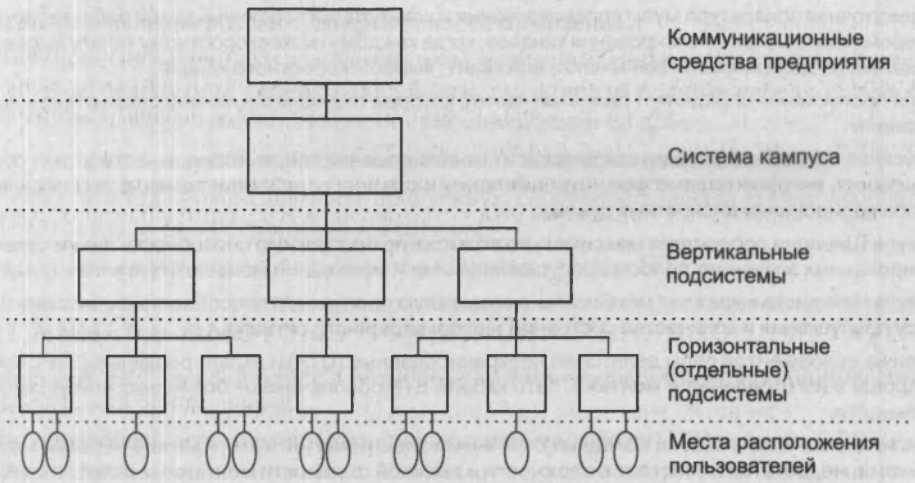


Рис. 8.18. Иерархия структурированной кабельной системы

Рис. 8.19. Структура кабельных подсистем

Кроме того, применение SCS делает *более экономичным* добавление новых пользователей и изменения их мест размещения. Известно, что стоимость кабельной системы определяется в основном не стоимостью кабеля, а стоимостью работ по его прокладке. Поэтому выгоднее провести однократную работу по прокладке кабеля, возможно, с большим запасом по длине, чем несколько раз выполнять прокладку, наращивая длину кабеля.

Выводы

В зависимости от типа промежуточной аппаратуры все линии связи делятся на аналоговые и цифровые. В аналоговых линиях промежуточная аппаратура предназначена для усиления аналоговых сигналов. В аналоговых линиях используется частотное мультиплексирование.

В цифровых линиях связи передаваемые сигналы имеют конечное число состояний. В таких линиях используется специальная промежуточная аппаратура — регенераторы, которые улучшают форму импульсов и обеспечивают их ресинхронизацию, то есть восстанавливают период их следования.

Промежуточная аппаратура мультиплексирования и коммутации первичных сетей работает по принципу временного мультиплексирования каналов, когда каждому низкоскоростному каналу выделяется определенная доля времени (тайм-слот, или квант) высокоскоростного канала.

Полоса пропускания определяет диапазон частот, которые передаются линией связи с приемлемым затуханием.

Пропускная способность линии связи зависит от ее внутренних параметров, в частности — полосы пропускания, внешних параметров — уровня помех и степени ослабления помех, а также принятого способа кодирования дискретных данных.

Формула Шеннона определяет максимально возможную пропускную способность линии связи при фиксированных значениях полосы пропускания линии и отношении мощности сигнала к шуму.

Формула Найквиста выражает максимально возможную пропускную способность линии связи через полосу пропускания и количество состояний информационного сигнала.

Кабели на основе витой пары делятся на неэкранированные (UTP) и экранированные (STP). Кабели UTP проще в изготовлении и монтаже, зато кабели STP обеспечивают более высокий уровень защищенности.

Волоконно-оптические кабели обладают отличными электромагнитными и механическими характеристиками, недостаток их состоит в сложности и высокой стоимости монтажных работ.

Структурированная кабельная система представляет собой набор коммуникационных элементов — кабелей, разъемов, коннекторов, кроссовых панелей и шкафов, которые удовлетворяют стандартам и позволяют создавать регулярные, легко расширяемые структуры связей.

Вопросы и задания

1. Синонимом каких терминов является термин «линия связи»? Варианты ответов:

   а) звено; б) канал; в) составной канал.
2. Назовите два основных типа среды передачи данных.
3. Может ли цифровой канал передавать аналоговые данные?
4. Чем отличаются усилители и регенераторы телекоммуникационных сетей?
5. Какими способами можно найти спектр сигнала?
6. Какое из окон прозрачности оптического волокна имеет наименьшее затухание? Варианты ответов:

   а) 850 нм; б) 1300 нм; в) 1550 нм.
7. Какие меры можно предпринять для увеличения информационной скорости звена? Варианты ответов:

   а) уменьшить длину кабеля;

   б) выбрать кабель с меньшим сопротивлением;

   в) выбрать кабель с более широкой полосой пропускания;

   г) применить метод кодирования с более узким спектром.
8. Чем отличается опорная мощность от относительной мощности? Варианты ответов:

   а) единицей измерения;

   б) фиксированной величиной мощности, к которой вычисляется отношение;

   в) длиной кабеля, на котором измеряется входная и выходная мощность;

9. Дайте определение порога чувствительности приемника.
10. Проверьте, достаточна ли для устойчивой передачи данных мощность передатчика в 40 дБм, если длина кабеля равна 60 км, погонное затухание кабеля составляет 0,2 дБ/км, а порог чувствительности приемника равен 20 дБм.
11. Что является причиной перекрестных наводок на ближнем конце кабеля?
12. Почему не всегда можно повысить пропускную способность канала за счет увеличения числа состояний информационного сигнала?
13. За счет какого механизма подавляются помехи в кабелях UTP?
14. Какой кабель более качественно передает сигналы, с большим значением параметра NEXT или с меньшим?
15. Какой тип кабеля предназначен для передачи данных на большие расстояния: многомодовый или одномодовый?
16. Что произойдет, если в работающей сети заменить кабель UTP кабелем STP? Варианты ответов:

    а) в сети снизится доля искаженных кадров;

    б) ничего не изменится;

    в) в сети увеличится доля искаженных кадров.
17. Каким будет теоретический предел скорости передачи данных в битах в секунду по линии связи с шириной полосы пропускания 1 мГц, если мощность передатчика составляет 64 дБм, а мощность шума в линии связи равна 2 дБм?

ГЛАВА 9 Кодирование и мультиплексирование данных

Проводные среды, которые мы рассмотрели в предыдущей главе, предоставляют только потенциальную возможность передачи дискретной информации. Для того чтобы передатчик и приемник, соединенные некоторой средой, могли обмениваться информацией, им необходимо договориться о том, какие сигналы будут соответствовать двоичным единицам и нулям дискретной информации. Для представления дискретной информации в среде передачи данных применяются сигналы двух типов: прямоугольные импульсы и синусоидальные волны. В первом случае используют термин «кодирование», во втором — «модуляция».

Существует множество способов кодирования, которые отличаются шириной спектра сигнала при одной и той же скорости передачи данных. Для передачи данных с минимальным числом ошибок полоса пропускания канала должна быть шире, чем спектр сигнала — иначе выбранные для представления единиц и нулей сигналы значительно исказятся, и приемник не сможет правильно распознать переданную информацию. Поэтому спектр сигнала является одним из главных критериев оценки эффективности способа кодирования.

Кроме того, способ кодирования должен способствовать синхронизации приемника с передатчиком, а также обеспечивать приемлемое соотношение мощности сигнала к шуму. Эти требования являются взаимно противоречивыми, поэтому каждый применяемый на практике способ кодирования представляет собой компромисс между основными требованиями.

Битовые ошибки в каналах связи нельзя исключить полностью, даже если выбранный код обеспечивает хорошую степень синхронизации и высокий уровень отношения сигнала к шуму. Поэтому при передаче дискретной информации применяются специальные коды, которые позволяют обнаруживать (а иногда даже исправлять) битовые ошибки.

Завершает главу рассмотрение методов мультиплексирования, которые позволяют образовать в одной линии связи несколько каналов передачи.

Модуляция

Модуляция при передаче аналоговых сигналов

Исторически модуляция начала применяться для *аналоговой информации* и только потом для дискретной.

Необходимость в модуляции аналоговой информации возникает, когда нужно передать низкочастотный аналоговый сигнал через канал, находящийся в высокочастотной области спектра. Примерами такой ситуация является передача голоса по радио или телевидению. Голос имеет спектр шириной примерно в 10 кГц, а радиодиапазоны включают гораздо более высокие частоты, от 30 кГц до 300 мГц. Еще более высокие частоты используются в телевидении. Очевидно, что непосредственно голос через такую среду передать нельзя.

Для решения проблемы амплитуду высокочастотного несущего сигнала изменяют (модулируют) в соответствии с изменением низкочастотного голосового сигнала (рис. 9.1). При этом спектр результирующего сигнала попадает в нужный высокочастотный диапазон. Такой тип модуляции называется **амплитудной модуляцией** (Amplitude Modulation, AM).

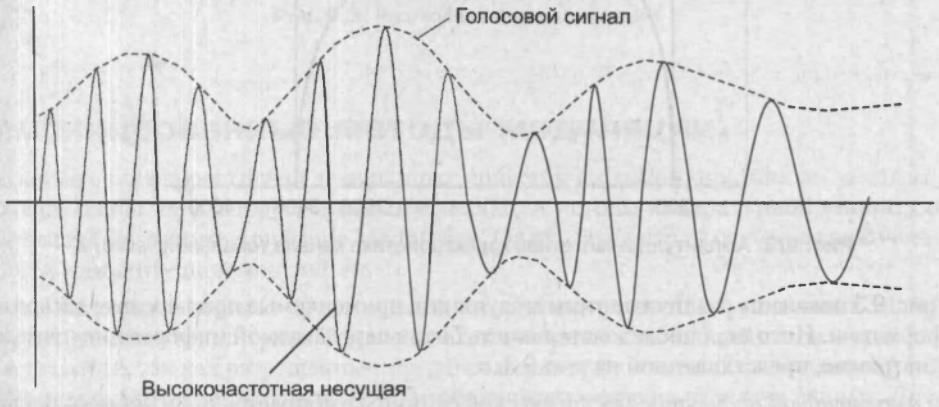


Рис. 9.1. Модуляция голосовым сигналом

В качестве информационного параметра используют не только амплитуду несущего синусоидального сигнала, но частоту. В этих случаях мы имеем дело с **частотной модуляцией** (Frequency Modulation, FM)[1].

Модуляция при передаче дискретных сигналов

При передаче *дискретной информации* посредством модуляции единицы и нули кодируются изменением амплитуды, частоты или фазы несущего синусоидального сигнала. В случае, когда модулированные сигналы передают дискретную информацию, вместо термина «модуляция» иногда используется термин «манипуляция»: амплитудная манипуляция (Amplitude Shift Keying, ASK), частотная манипуляция (Frequency Shift Keying, FSK), фазовая манипуляция (Phase Shift Keying, PSK).

---

[1] Заметим, что при модуляции аналоговой информации фаза как информационный параметр не применяется.

Пожалуй, самый известный пример применения модуляции при передаче дискретной информации — это передача компьютерных данных по телефонным каналам. Типичная амплитудно-частотная характеристика стандартного абонентского канала, называемого также **каналом тональной частоты**, представлена на рис. 9.2. Этот составной канал проходит через коммутаторы телефонной сети и соединяет телефоны абонентов. Канал тональной частоты передает частоты в диапазоне от 300 до 3400 Гц, таким образом, его полоса пропускания равна 3100 Гц. Такая узкая полоса пропускания вполне достаточна для качественной передачи голоса, однако она недостаточно широка для передачи компьютерных данных в виде прямоугольных импульсов. Решение проблемы было найдено благодаря аналоговой модуляции. Устройство, которое выполняет функцию *модуляции* несущей синусоиды на передающей стороне и обратную функцию *демодуляции* на приемной стороне, носит название **модема** (**модулятор-демодулятор**).

Рис. 9.2. Амплитудно-частотная характеристика канала тональной частоты

На рис. 9.3 показаны различные типы модуляции, применяемые при передаче дискретной информации. Исходная последовательность битов передаваемой информации приведена на диаграмме, представленной на рис. 9.3, *а*.

При *амплитудной модуляции* для логической единицы выбирается один уровень амплитуды синусоиды несущей частоты, а для логического нуля — другой (рис. 9.3, *б*). Этот способ редко используется в чистом виде на практике из-за низкой помехоустойчивости, но часто применяется в сочетании с другим видом модуляции — фазовой модуляцией.

При *частотной модуляции* значения нуля и единицы исходных данных передаются синусоидами с различной частотой — f_0 и f_1 (рис. 9.3, *в*). Этот способ модуляции не требует сложных схем и обычно применяется в низкоскоростных модемах, работающих на скоростях 300 и 1200 бит/с. При использовании только двух частот за один такт передается один бит информации, поэтому такой способ называется **двоичной частотной манипуляцией** (Binary FSK, BFSK). Могут также использоваться четыре различные частоты для кодирования двух битов информации в одном такте, такой способ носит название **четырехуровневой частотной манипуляции** (four-level FSK). Применяется также название **многоуровневая частотная манипуляция** (Multilevel FSK, MFSK).

При **фазовой модуляции** значениям данных 0 и 1 соответствуют сигналы одинаковой частоты, но различной фазы, например 0 и 180° или 0, 90, 180 и 270° (рис. 9.3, *г*). В первом случае такая модуляция носит название **двоичной фазовой манипуляции** (Binary PSK, BPSK), а во втором — **квадратурной фазовой манипуляции** (Quadrature PSK, QPSK).

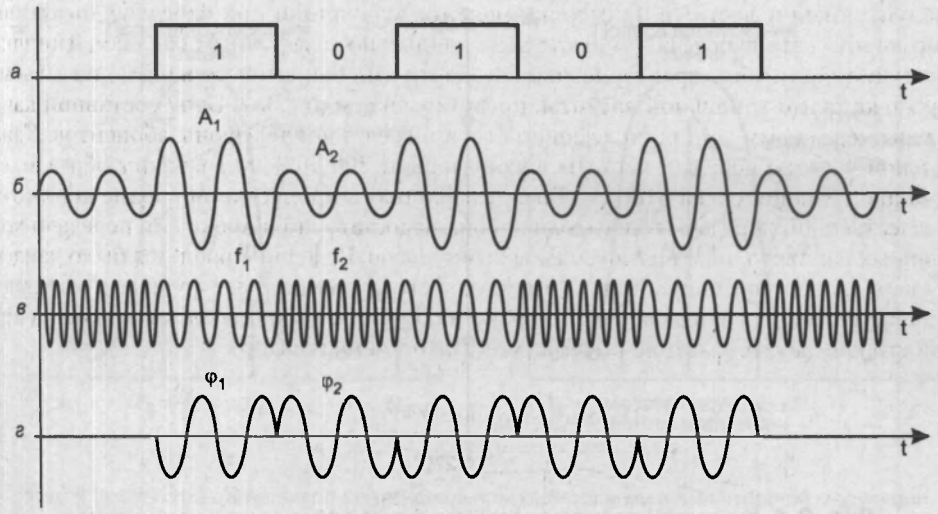


Рис. 9.3. Различные типы модуляции

Комбинированные методы модуляции

Для повышения скорости передачи данных прибегают к комбинированным методам модуляции. Наиболее распространенными являются методы **квадратурной амплитудной модуляции** (Quadrature Amplitude Modulation, QAM). Эти методы основаны на сочетании фазовой и амплитудной модуляции.

На рис. 9.4 показан вариант модуляции, в котором используется 8 различных значений фазы и 4 значения амплитуды. Однако из 32 возможных комбинаций сигнала задействовано только 16, так как разрешенные значения амплитуд у соседних фаз отличаются. Это повышает помехоустойчивость кода, но вдвое снижает скорость передачи данных. Другим решением, повышающим надежность кода за счет введения избыточности, являются так называемые **решетчатые коды**. В этих кодах к каждым четырем битам информации добавляется пятый бит, который даже при наличии ошибок позволяет с большой степенью вероятности определить правильный набор четырех информационных битов.

Спектр результирующего модулированного сигнала зависит от *типа модуляции* и скорости модуляции, то есть желаемой *скорости передачи* битов исходной информации.

Рассмотрим сначала спектр сигнала при потенциальном кодировании. Пусть логическая единица кодируется положительным потенциалом, а логический ноль — отрицательным потенциалом такой же величины. Для упрощения вычислений предположим, что передается информация, состоящая из бесконечной последовательности чередующихся единиц и нулей, как показано на рис. 9.3, *а*.

Спектр непосредственно получается из формул Фурье для периодической функции. Если дискретные данные передаются с битовой скоростью N бит/с, то спектр состоит из постоянной составляющей нулевой частоты и бесконечного ряда гармоник с частотами f_0, $3f_0$, $5f_0$, $7f_0$, ..., где $f_0 = N/2$. Частота f_0 — первая частота спектра — называется **основной гармоникой**.

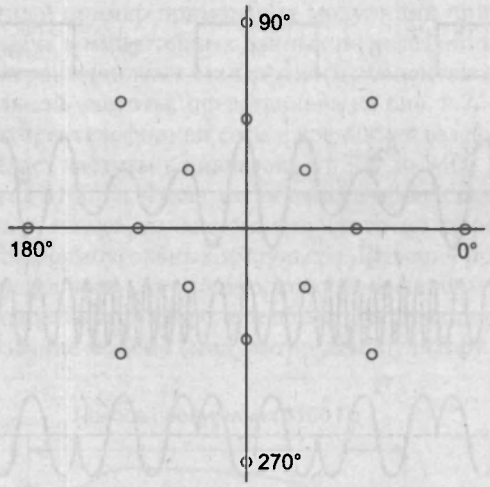


Рис. 9.4. Квадратурная амплитудная модуляция с 16-ю состояниями сигнала

Амплитуды этих гармоник убывают достаточно медленно — с коэффициентами 1/3, 1/5, 1/7, ... от амплитуды гармоники f_0 (рис. 9.5, *а*). В результате спектр потенциального кода требует для качественной передачи широкую полосу пропускания. Кроме того, нужно учесть, что реально спектр сигнала постоянно меняется в зависимости от того, какие данные передаются по линии связи. Например, передача длинной последовательности нулей или единиц сдвигает спектр в сторону низких частот, а в крайнем случае, когда передаваемые данные состоят только из единиц (или только из нулей), спектр состоит из гармоники нулевой частоты. При передаче чередующихся единиц и нулей постоянная составляющая отсутствует. Поэтому спектр результирующего сигнала потенциального кода при передаче произвольных данных занимает полосу от некоторой величины, близкой к нулю, до примерно $7f_0$ (гармониками с частотами выше $7f_0$ можно пренебречь из-за их малого вклада в результирующий сигнал). Для канала тональной частоты верхняя граница при потенциальном кодировании достигается для скорости передачи данных в 971 бит/с, а нижняя неприемлема для любых скоростей, так как полоса пропускания канала начинается с 300 Гц. В результате потенциальные коды на каналах тональной частоты никогда не используются.

При амплитудной модуляции спектр состоит из синусоиды несущей частоты f_c, двух боковых гармоник ($f_c + f_m$) и ($f_c - f_m$), а также боковых гармоник ($f_c + 3f_m$) и ($f_c - 3f_m$), где f_m — частота изменения информационного параметра синусоиды, которая совпадает со скоростью передачи данных при использовании двух уровней амплитуды (рис. 9.5, *б*). Частота f_m определяет пропускную способность линии при данном способе кодирования. На небольшой частоте модуляции ширина спектра сигнала также оказывается небольшой (равной $2f_m$), если пренебречь гармониками $3f_m$, мощность которых незначительна.

При фазовой и частотной модуляции спектр сигнала получается более сложным, чем при амплитудной модуляции, так как боковых гармоник здесь образуется более двух, но они тоже симметрично расположены относительно основной несущей частоты, а их амплитуды быстро убывают.

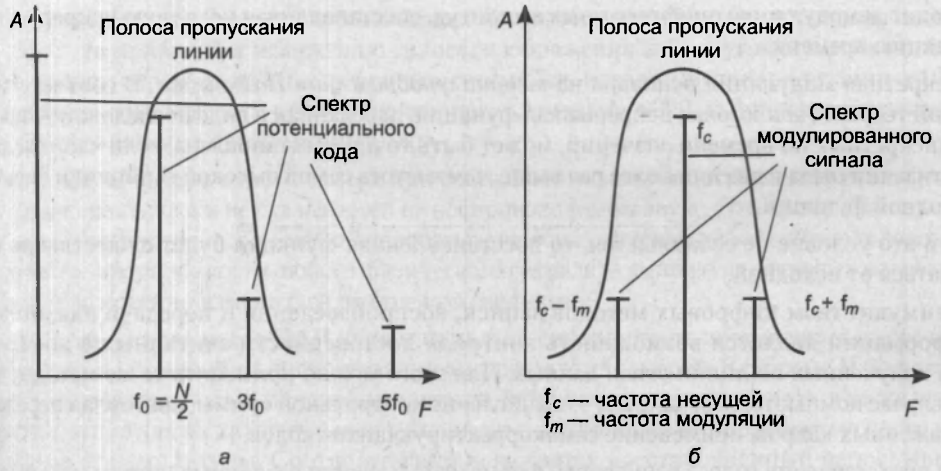


Рис. 9.5. Спектры сигналов при потенциальном кодировании и амплитудной модуляции

Дискретизация аналоговых сигналов

В предыдущем разделе мы познакомились с преобразованием дискретной формы представления информации в аналоговую. В этом разделе рассматривается решение обратной задачи — передачи аналоговой информации в дискретной форме.

Как мы уже упоминали в главе 3, начиная с 60-х годов прошлого века голос начал передаваться по телефонным сетям в цифровой форме, то есть в виде последовательности единиц и нулей. Основной причиной такого перехода является невозможность улучшения качества данных, переданных в аналоговой форме, если они существенно исказились при передаче. Сам аналоговый сигнал не дает никаких указаний ни на то, что произошло искажение, ни на то, как его исправить, поскольку форма сигнала может быть любой, в том числе и такой, которую зафиксировал приемник. Улучшение же качества линий, особенно территориальных, требует огромных усилий и капиталовложений. Поэтому на смену аналоговой технике записи и передачи звука и изображений пришла цифровая техника. В этой технике используется так называемая **дискретная модуляция** исходных непрерывных во времени аналоговых процессов.

Амплитуда исходной непрерывной функции измеряется с заданным периодом — за счет этого происходит *дискретизация по времени*.

Затем каждый замер представляется в виде двоичного числа определенной разрядности, что означает *дискретизацию по значениям* — непрерывное множество возможных значений амплитуды заменяется дискретным множеством ее значений.

Устройство, которое выполняет подобную функцию, называется **аналого-цифровым преобразователем** (АЦП). После этого замеры передаются по линиям связи в виде последовательности единиц и нулей. При этом применяются те же методы кодирования (с ними мы познакомимся позднее), что и при передаче изначально дискретной информации.

На приемной стороне линии коды преобразуются в исходную последовательность битов, а специальная аппаратура, называемая **цифро-аналоговым преобразователем** (ЦАП), про-

изводит демодуляцию оцифрованных амплитуд, восстанавливая исходную непрерывную функцию времени.

Дискретная модуляция основана на *теории отображения Найквиста*. В соответствии с этой теорией, аналоговая непрерывная функция, переданная в виде последовательности ее дискретных по времени значений, может быть точно восстановлена, если частота дискретизации была в два или более раз выше, чем частота самой высокой гармоники спектра исходной функции.

Если это условие не соблюдается, то восстановленная функция будет существенно отличаться от исходной.

Преимуществом цифровых методов записи, воспроизведения и передачи аналоговой информации является возможность контроля достоверности считанных с носителя или полученных по линии связи данных. Для этого можно применять те же методы, что и в случае компьютерных данных, — вычисление контрольной суммы, повторная передача искаженных кадров, применение самокорректирующихся кодов.

Для представления голоса в цифровой форме используются различные методы его дискретизации. Наиболее простой метод, в котором применяется частота квантования амплитуды звуковых колебаний в 8000 Гц, уже был кратко рассмотрен в главе 3. Этот метод имеет название **импульсно-кодовой модуляции** (Pulse Code Modulation, PCM).

Обоснование выбранной частоты квантования в методе PCM достаточно простое. Оно объясняется тем, что в аналоговой телефонии для передачи голоса был выбран диапазон от 300 до 3400 Гц, который достаточно качественно передает все основные гармоники собеседников. В соответствии с *теоремой Найквиста—Котельникова* для качественной передачи голоса достаточно выбрать частоту дискретизации, в два раза превышающую самую высокую гармонику непрерывного сигнала, то есть $2 \times 3400 = 6800$ Гц. Выбранная в действительности частота дискретизации 8000 Гц обеспечивает некоторый запас качества. В методе PCM обычно используется 7 или 8 бит кода для представления амплитуды одного замера. Соответственно это дает 127 или 256 градаций звукового сигнала, что оказывается вполне достаточно для качественной передачи голоса.

При использовании метода PCM для передачи одного голосового канала необходима пропускная способность 56 или 64 Кбит/с в зависимости от того, каким количеством битов представляется каждый замер. Если для этих целей применяется 7 бит, то при частоте передачи замеров в 8000 Гц получаем:

$$8000 \times 7 = 56\,000 \text{ бит/с или } 56 \text{ Кбит/с.}$$

А для случая 8 бит:

$$8000 \times 8 = 64\,000 \text{ бит/с или } 64 \text{ Кбит/с.}$$

Как вы знаете, стандартным является цифровой канал 64 Кбит/с, который также называется **элементарным каналом цифровых телефонных сетей**; канал 56 Кбит/с применялся на ранних этапах существования цифровой телефонии, когда один бит из байта, отведенного для передачи данных, изымался для передачи номера вызываемого абонента (детали см. в разделе «Сети PDH» главы 11).

Передача непрерывного сигнала в дискретном виде требует от сетей жесткого соблюдения временного интервала в 125 мкс (соответствующего частоте дискретизации 8000 Гц) между соседними замерами, то есть требует синхронной передачи данных между узлами сети.

При отсутствии синхронности прибывающих замеров исходный сигнал восстанавливается неверно, что приводит к искажению голоса, изображения или другой мультимедийной информации, передаваемой по цифровым сетям. Так, искажение синхронизации в 10 мс может привести к эффекту «эха», а сдвиги между замерами в 200 мс приводят к невозможности распознавания произносимых слов.

В то же время потеря одного замера при соблюдении синхронности между остальными замерами практически не сказывается на воспроизводимом звуке. Это происходит за счет сглаживающих устройств в цифро-аналоговых преобразователях, работа которых основана на свойстве инерционности любого физического сигнала — амплитуда звуковых колебаний не может мгновенно измениться на большую величину.

На качество сигнала после ЦАП влияет не только синхронность поступления на его вход замеров, но и погрешность дискретизации амплитуд этих замеров. В теореме Найквиста—Котельникова предполагается, что амплитуды функции измеряются точно, в то же время использование для их хранения двоичных чисел с ограниченной разрядностью несколько искажает эти амплитуды. Соответственно искажается восстановленный непрерывный сигнал – этот эффект называют шумом дискретизации (по амплитуде).

Методы кодирования

Выбор способа кодирования

При выборе способа кодирования нужно одновременно стремиться к достижению нескольких целей:

- минимизировать ширину спектра сигнала, полученного в результате кодирования;
- обеспечивать синхронизацию между передатчиком и приемником;
- обеспечивать устойчивость к шумам;
- обнаруживать и по возможности исправлять битовые ошибки;
- минимизировать мощность передатчика.

Более узкий **спектр сигнала** позволяет на одной и той же линии (с одной и той же полосой пропускания) добиваться более высокой скорости передачи данных. Спектр сигнала в общем случае зависит как от способа кодирования, так и от тактовой частоты передатчика. Пусть мы разработали два способа кодирования, причем в каждом такте передается один бит информации. Пусть также в первом способе ширина спектра сигнала F равна тактовой частоте смены сигналов f, то есть $F = f$, а второй способ дает зависимость $F = 0{,}8f$. Тогда при одной и той же полосе пропускания B первый способ позволит передавать данные со скоростью B бит/с, а второй $(1/0{,}8)B = 1{,}25\,B$ бит/с.

Синхронизация передатчика и приемника нужна для того, чтобы приемник точно знал, в какой момент времени считывать новую порцию информации с линии связи. При передаче дискретной информации время всегда разбивается на такты одинаковой длительности, и приемник старается считать новый сигнал в середине каждого такта, то есть синхронизировать свои действия с передатчиком.

Проблема синхронизации в сетях решается сложнее, чем при обмене данными между близко расположенными устройствами, например между блоками внутри компьютера

или же между компьютером и принтером. На небольших расстояниях хорошо работает схема, основанная на отдельной *тактирующей линии связи* (рис. 9.6), так что информация снимается с линии данных только в момент прихода тактового импульса. В сетях использование этой схемы вызывает трудности из-за неоднородности характеристик проводников в кабелях. На больших расстояниях неравномерность скорости распространения сигнала может привести к тому, что тактовый импульс придет настолько позже или раньше соответствующего сигнала данных, что бит данных будет пропущен или считан повторно. Другой причиной, по которой в сетях отказываются от использования тактирующих импульсов, является экономия проводников в дорогостоящих кабелях.

Рис. 9.6. Синхронизация приемника и передатчика на небольших расстояниях

В сетях для решения проблемы синхронизации применяются так называемые **самосинхронизирующиеся коды**, сигналы которых несут для приемника указания о том, в какой момент времени начать распознавание очередного бита (или нескольких битов, если код ориентирован более чем на два состояния сигнала). Любой резкий перепад сигнала — **фронт** — может служить указанием на необходимость синхронизации приемника с передатчиком.

При использовании синусоид в качестве несущего сигнала результирующий код обладает свойством самосинхронизации, так как изменение амплитуды несущей частоты дает возможность приемнику определить момент очередного такта.

Распознавание и коррекцию искаженных данных сложно осуществить средствами физического уровня, поэтому чаще всего эту работу берут на себя протоколы, лежащие выше: канальный, сетевой, транспортный или прикладной. В то же время распознавание ошибок на физическом уровне экономит время, так как приемник не ждет полного помещения кадра в буфер, а отбраковывает его сразу при распознавании ошибочных битов внутри кадра.

Требования, предъявляемые к методам кодирования, являются взаимно противоречивыми, поэтому каждый из рассматриваемых далее популярных методов кодирования обладает своими достоинствами и недостатками в сравнении с другими.

Потенциальный код NRZ

Рисунок 9.7, *а* иллюстрирует уже упомянутый ранее метод *потенциального кодирования*, называемый также кодированием **без возвращения к нулю** (Non Return to Zero, NRZ). Последнее название отражает то обстоятельство, что в отличие от других методов кодирования при передаче последовательности единиц сигнал не возвращается к нулю в течение такта.

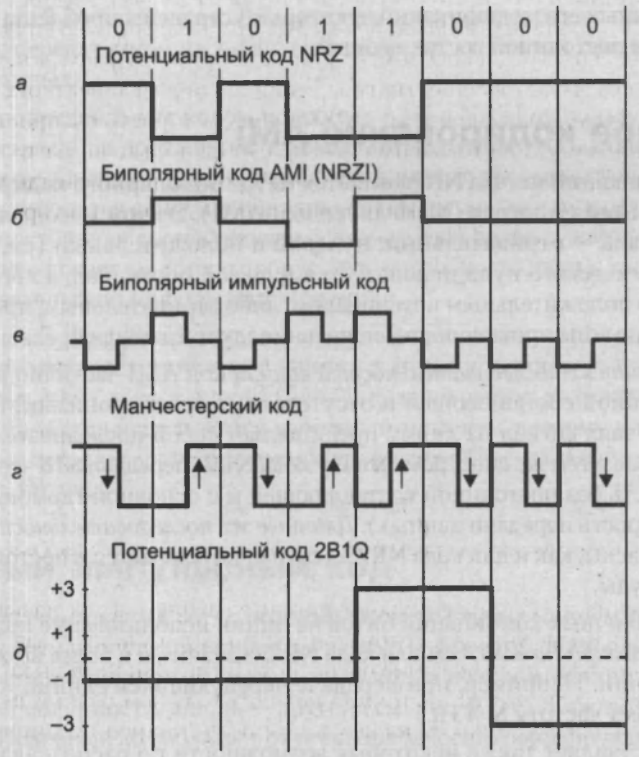


Рис. 9.7. Способы дискретного кодирования данных

Итак, достоинства метода NRZ.

- Простота реализации.
- Метод обладает хорошей распознаваемостью ошибок (благодаря наличию двух резко отличающихся потенциалов).
- Основная гармоника f_0 имеет достаточно низкую частоту (равную $N/2$ Гц, как было показано в предыдущем разделе), что приводит к узкому спектру.

Теперь недостатки метода NRZ.

- Метод не обладает свойством самосинхронизации. Даже при наличии высокоточного тактового генератора приемник может ошибиться с выбором момента съема данных, так как частоты двух генераторов никогда не бывают полностью идентичными. Поэтому при высоких скоростях обмена данными и длинных последовательностях единиц или нулей небольшое рассогласование тактовых частот может привести к ошибке в целый такт и, соответственно, считыванию некорректного значения бита.
- Вторым серьезным недостатком метода NRZ является наличие низкочастотной составляющей, которая приближается к постоянному сигналу при передаче длинных последовательностей единиц или нулей. Из-за этого многие линии связи, не обеспечивающие прямого гальванического соединения между приемником и источником, этот вид кодирования не поддерживают. Поэтому в сетях код NRZ в основном используется

в виде различных его модификаций, в которых устранены проблемы плохой самосинхронизации и постоянной составляющей.

Биполярное кодирование AMI

Одной из модификаций метода NRZ является метод **биполярного кодирования с альтернативной инверсией** (Alternate Mark Inversion, AMI). В этом методе применяются три уровня потенциала — отрицательный, нулевой и положительный (см. рис. 9.7, *б*). Для кодирования логического нуля используется нулевой потенциал, а логическая единица кодируется либо положительным потенциалом, либо отрицательным, при этом потенциал каждой новой единицы противоположен потенциалу предыдущей.

При передаче *длинных последовательностей единиц* код AMI частично решает проблемы наличия постоянной составляющей и отсутствия самосинхронизации, присущие коду NRZ. В этих случаях сигнал на линии представляет собой последовательность разнополярных импульсов с тем же спектром, что и у кода NRZ, передающего чередующиеся нули и единицы, то есть без постоянной составляющей и с основной гармоникой $N/2$ Гц (где N — битовая скорость передачи данных). *Длинные же последовательности нулей* для кода AMI столь же опасны, как и для кода NRZ — сигнал вырождается в постоянный потенциал нулевой амплитуды.

В целом, для различных комбинаций битов на линии использование кода AMI приводит к *более узкому спектру сигнала*, чем для кода NRZ, а значит, и к более высокой пропускной способности линии. Например, при передаче чередующихся единиц и нулей основная гармоника f_0 имеет частоту $N/4$ Гц.

Код AMI предоставляет также некоторые возможности по распознаванию ошибочных сигналов. Так, нарушение строгой очередности в полярности сигналов говорит о ложном импульсе или исчезновении с линии корректного импульса.

В коде AMI используются не два, а три уровня сигнала на линии. Дополнительный уровень требует увеличение мощности передатчика примерно на 3 дБ для обеспечения той же достоверности приема битов на линии, что является общим недостатком кодов с несколькими состояниями сигнала по сравнению с кодами, в которых различают только два состояния.

Потенциальный код NRZI

Существует код, похожий на AMI, но только с двумя уровнями сигнала. При передаче нуля он передает потенциал, который был установлен на предыдущем такте (то есть не меняет его), а при передаче единицы потенциал инвертируется на противоположный. Этот код называется **потенциальным кодом с инверсией при единице** (Non Return to Zero with ones Inverted, NRZI). Он удобен в тех случаях, когда наличие третьего уровня сигнала весьма нежелательно, например в оптических кабелях, где устойчиво распознаются только два состояния сигнала — свет и темнота.

Код NRZI хорош тем, что в среднем требует меньше изменений сигнала при передаче произвольной двоичной информации, чем манчестерский код, за счет чего спектр его сигналов уже. Однако код NRZI обладает плохой самосинхронизацией, так как при передаче длинных последовательностей нулей сигнал вообще не меняется (например, при

передаче последних 3-х нулей на рис. 9.7, *а*), и, значит, у приемника исчезает возможность синхронизации с передатчиком на значительное время, что может приводить к ошибкам распознавания данных.

Для улучшения потенциальных кодов, подобных AMI и NRZI, используются два метода. Первый метод основан на добавлении в исходный код избыточных битов, содержащих логические единицы. Очевидно, что в этом случае длинные последовательности нулей прерываются, и код становится самосинхронизирующимся для любых передаваемых данных. Исчезает также постоянная составляющая, а значит, еще более сужается спектр сигнала. Однако этот метод снижает полезную пропускную способность линии, так как избыточные единицы пользовательской информации не несут.

Другой метод основан на предварительном «перемешивании» исходной информации таким образом, чтобы вероятность появления единиц и нулей на линии становилась близкой к нулю. Устройства, или блоки, выполняющие такую операцию, называются **скрэмблерами**. При скремблировании используется известный алгоритм, поэтому приемник, получив двоичные данные, передает их на **дескрэмблер**, который восстанавливает исходную последовательность битов.

Биполярный импульсный код

Помимо *потенциальных кодов* в сетях используются и *импульсные коды*, в которых данные представлены полным импульсом или же его частью — фронтом. Наиболее простым кодом такого рода является **биполярный импульсный код**, в котором единица представляется импульсом одной полярности, а ноль — другой (см. рис. 9.7, *в*). Каждый импульс длится половину такта. Подобный код обладает отличными самосинхронизирующими свойствами, но постоянная составляющая может присутствовать, например, при передаче длинной последовательности единиц или нулей. Кроме того, спектр у него шире, чем у потенциальных кодов. Так, при передаче всех нулей или единиц частота основной гармоники кода равна *N* Гц, что в два раза выше основной гармоники кода NRZ и в четыре раза выше основной гармоники кода AMI при передаче чередующихся единиц и нулей. Из-за слишком широкого спектра биполярный импульсный код используется редко.

Манчестерский код

В локальных сетях до недавнего времени самым распространенным был так называемый **манчестерский код** (см. рис. 9.7, *г*). Он применяется в технологиях Ethernet и Token Ring.

В манчестерском коде для кодирования единиц и нулей используется перепад потенциала, то есть фронт импульса. При манчестерском кодировании каждый такт делится на две части. Информация кодируется перепадами потенциала, происходящими в середине каждого такта. Единица кодируется перепадом от низкого уровня сигнала к высокому, а ноль — обратным перепадом. В начале каждого такта может происходить служебный перепад сигнала, если нужно представить несколько единиц или нулей подряд. Так как сигнал изменяется, по крайней мере, один раз за такт передачи одного бита данных, то манчестерский код обладает хорошими самосинхронизирующими свойствами. Полоса пропускания манчестерского кода уже, чем у биполярного импульсного. У него также нет постоянной составляющей, к тому же основная гармоника в худшем случае (при передаче

последовательности единиц или нулей) имеет частоту N Гц, а в лучшем (при передаче чередующихся единиц и нулей) — $N/2$ Гц, как и у кодов AMI и NRZ. В среднем ширина полосы манчестерского кода в полтора раза уже, чем у биполярного импульсного кода, а основная гармоника колеблется вблизи значения $3N/4$. Манчестерский код имеет еще одно преимущество перед биполярным импульсным кодом. В последнем для передачи данных используются три уровня сигнала, а в манчестерском — два.

Потенциальный код 2B1Q

На рис. 9.7, *д* показан потенциальный код с четырьмя уровнями сигнала для кодирования данных. Это код **2B1Q**, название которого отражает его суть — каждые два бита (2B) передаются за один такт (1) сигналом, имеющим четыре состояния (Q — Quadra). Паре битов 00 соответствует потенциал –2,5 В, паре 01 — потенциал –0,833 В, паре 11 — потенциал +0,833 В, а паре 10 — потенциал +2,5 В.

При этом способе кодирования требуются дополнительные меры по борьбе с длинными последовательностями одинаковых пар битов, так как при этом сигнал превращается в постоянную составляющую. При случайном чередовании битов спектр сигнала в два раза уже, чем у кода NRZ, так как при той же битовой скорости длительность такта увеличивается в два раза. Таким образом, с помощью кода 2B1Q можно по одной и той же линии передавать данные в два раза быстрее, чем с помощью кода AMI или NRZI. Однако для его реализации мощность передатчика должна быть выше, чтобы четыре уровня четко различались приемником на фоне помех.

Для улучшения потенциальных кодов типа AMI, NRZI или 2Q1B используются избыточные коды и скрэмблирование.

Избыточный код 4B/5B

Избыточные коды основаны на разбиении исходной последовательности битов на порции, которые часто называют *символами*. Затем каждый исходный символ заменяется новым с бо́льшим количество битов, чем исходный.

Например, в логическом коде **4B/5B**, используемом в технологиях FDDI и Fast Ethernet, исходные символы длиной 4 бит заменяются символами длиной 5 бит. Так как результирующие символы содержат избыточные биты, то общее количество битовых комбинаций в них больше, чем в исходных. Так, в коде 4B/5B результирующие символы могут содержать 32 битовые комбинации, в то время как исходные символы — только 16 (табл. 9.1). Поэтому в результирующем коде можно отобрать 16 таких комбинаций, которые не содержат большого количества нулей, а остальные считать **запрещенными кодами** (code violations). Помимо устранения постоянной составляющей и придания коду свойства самосинхронизации, избыточные коды позволяют приемнику распознавать искаженные биты. Если приемник принимает запрещенный код, значит, на линии произошло искажение сигнала.

После разбиения получившийся код 4B/5B передается по линии путем преобразования с помощью какого-либо из методов потенциального кодирования, чувствительного только к длинным последовательностям нулей. Таким кодом является, например, код NRZI.

Символы кода 4B/5B длиной 5 бит гарантируют, что при любом их сочетании на линии не встретятся более трех нулей подряд.

Таблица 9.1. Соответствие исходных и результирующих кодов 4B/5B

Исходный код	Результирующий код	Исходный код	Результирующий код
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

ПРИМЕЧАНИЕ

Буква B в названии кода 4B/5B означает, что элементарный сигнал имеет два состояния (от английского binary — двоичный). Имеются также коды и с тремя состояниями сигнала, например в коде 8B/6T для кодирования 8 бит исходной информации используется код из 6 сигналов, каждый из которых имеет три состояния. Избыточность кода 8B/6T выше, чем кода 4B/5B, так как на 256 исходных кодов приходится $3^6 = 729$ результирующих символов.

Использование таблицы перекодировки является очень простой операцией, поэтому этот подход не усложняет сетевые адаптеры и интерфейсные блоки коммутаторов и маршрутизаторов.

Для обеспечения заданной пропускной способности линии передатчик, использующий избыточный код, должен работать с повышенной тактовой частотой. Так, для передачи кодов 4B/5B со скоростью 100 Мбит/с требуется тактовая частота 125 МГц. При этом спектр сигнала на линии расширяется по сравнению со случаем, когда по линии передается не избыточный код. Тем не менее спектр избыточного потенциального кода оказывается уже спектра манчестерского кода, что оправдывает дополнительный этап логического кодирования, а также работу приемника и передатчика на повышенной тактовой частоте.

Скремблирование

Скремблирование заключается в побитном вычислении результирующего кода на основании битов исходного кода и полученных в предыдущих тактах битов результирующего кода. Например, скрэмблер может реализовывать следующее соотношение:

$$B_i = A_i \, B_{i-3} \, B_{i-5}.$$

Здесь B_i — двоичная цифра результирующего кода, полученная на i-м такте работы скрэмблера, A_i — двоичная цифра исходного кода, поступающая на i-м такте на вход скрэмблера, B_{i-3} и B_{i-5} — двоичные цифры результирующего кода, полученные на предыдущих тактах работы скрэмблера (соответственно на 3 и на 5 тактов ранее текущего такта) и объединенные операцией исключающего ИЛИ (сложение по модулю 2).

Например, для исходной последовательности 110110000001 скрэмблер даст следующий результирующий код (первые три цифры результирующего кода будут совпадать с исходным кодом, так как еще нет нужных предыдущих цифр):

$B_1 = A_1 = 1$

$B_2 = A_2 = 1$

$B_3 = A_3 = 0$

$B_4 = A_4 B_1 = 1 1 = 0$

$B_5 = A_5 B_2 = 1 1 = 0$

$B_6 = A_6 B_3 B_1 = 0 0 1 = 1$

$B_7 = A_7 B_4 B_2 = 0 0 1 = 1$

$B_8 = A_8 B_5 B_3 = 0 0 0 = 0$

$B_9 = A_9 B_6 B_4 = 0 1 0 = 1$

$B_{10} = A_{10} B_7 B_5 = 0 1 0 = 1$

$B_{11} = A_{11} B_8 B_6 = 0 0 1 = 1$

$B_{12} = A_{12} B_9 B_7 = 1 1 1 = 1$

Таким образом, на выходе скрэмблера появится код 110001101111, в котором нет последовательности из шести нулей, присутствовавшей в исходном коде.

После получения результирующей последовательности приемник передает ее дескрэмблеру, который восстанавливает исходную последовательность на основании обратного соотношения:

$$C_i = B_i B_{i-3} B_{i-5} = (A_i B_{i-3} B_{i-5}) B_{i-3} B_{i-5} = A_i.$$

Различные алгоритмы скрэмблирования отличаются количеством слагаемых, дающих цифру результирующего кода, и сдвигом между слагаемыми. Так, в сетях ISDN при передаче данных от сети к абоненту используется преобразование со сдвигами на 5 и 23 позиции, а при передаче данных от абонента в сеть — со сдвигами на 18 и 23 позиции.

Существуют и более простые методы борьбы с последовательностями единиц, также относимые к классу скрэмблирования. Для улучшения биполярного кода AMI используются два метода, основанные на искусственном искажении последовательности нулей *запрещенными символами*.

Рисунок 9.8 иллюстрирует использование методов **B8ZS** (Bipolar with 8-Zeros Substitution) и **HDB3** (High-Density Bipolar 3-Zeros) для корректировки кода AMI. Исходный код состоит из двух длинных последовательностей нулей: в первом случае — из 8, а во втором — из 5.

Код B8ZS исправляет только последовательности, состоящие из 8 нулей. Для этого он после первых трех нулей вместо оставшихся пяти нулей вставляет пять цифр: *V*-1*-0-*V*-1*. Здесь *V* обозначает сигнал единицы, запрещенной (*V*iolations) для данного такта полярности, то есть сигнал, не изменяющий полярность предыдущей единицы, 1* — сигнал единицы корректной полярности (знак звездочки отмечает тот факт, что в исходном коде в этом такте была не единица, а ноль). В результате на 8 тактах приемник наблюдает 2 искажения — очень маловероятно, что это случается из-за шума на линии или других сбоев передачи. Поэтому приемник считает такие нарушения кодировкой 8 последовательных

нулей и после приема заменяет их исходными 8 нулями. Код B8ZS построен так, что его постоянная составляющая равна нулю при любых последовательностях двоичных цифр.

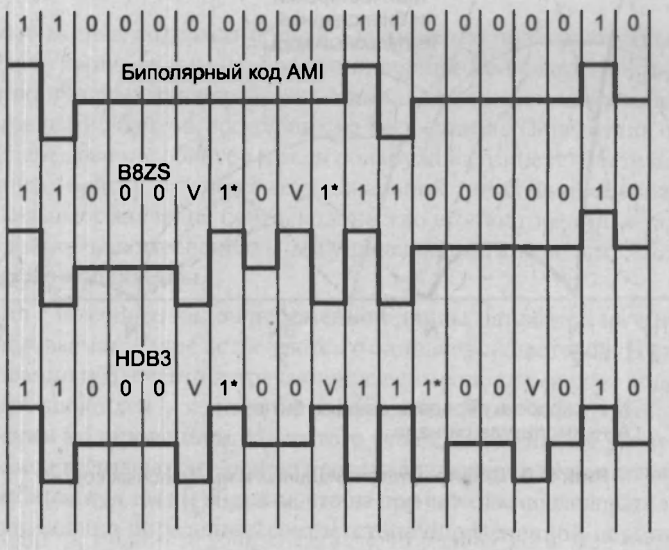
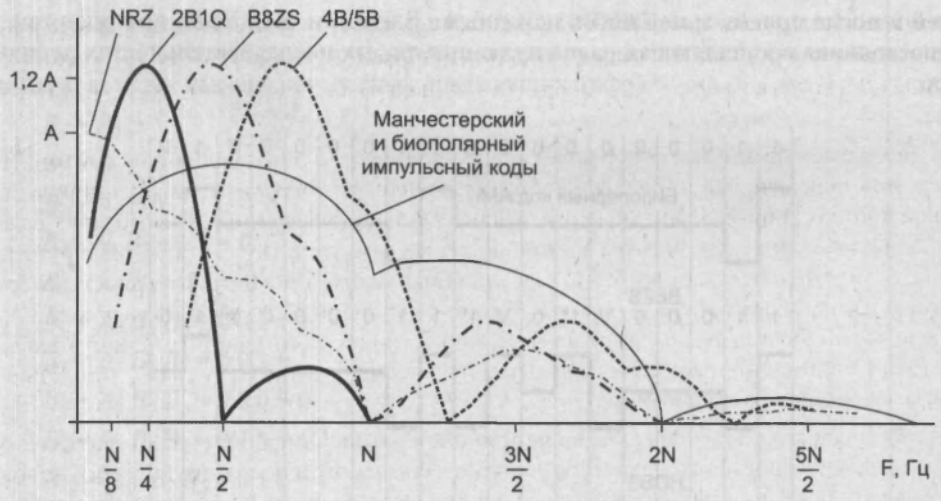


Рис. 9.8. Коды B8ZS и HDB3

Код HDB3 исправляет любые четыре смежных нуля в исходной последовательности. Правила формирования кода HDB3 более сложные, чем кода B8ZS. Каждые четыре нуля заменяются четырьмя сигналами, в которых имеется один сигнал *V*. Для подавления постоянной составляющей полярность сигнала *V* чередуется при последовательных заменах. Кроме того, для замены используются два образца четырехтактовых кодов. Если перед заменой исходный код содержал нечетное число единиц, задействуется последовательность 000*V*, а если число единиц было четным — последовательность 1*00*V*.

Улучшенные потенциальные коды обладают достаточно узкой полосой пропускания для любых последовательностей единиц и нулей, которые встречаются в передаваемых данных. На рис. 9.9 приведены спектры сигналов разных кодов, полученные при передаче произвольных данных, в которых различные сочетания нулей и единиц в исходном коде равновероятны. При построении графиков спектр усреднялся по всем возможным наборам исходных последовательностей. Естественно, что результирующие коды могут иметь и другое распределение нулей и единиц. Из рисунка видно, что потенциальный код NRZ обладает хорошим спектром с одним недостатком — у него имеется постоянная составляющая. Коды, полученные из потенциального кода путем логического кодирования, обладают более узким спектром, чем манчестерский код, даже при повышенной тактовой частоте (на рисунке спектр кода 4B/5B должен был бы примерно совпадать с кодом B8ZS, но он сдвинут в область более высоких частот, так как его тактовая частота повышена на 1/4 по сравнению с другими кодами). Этим объясняется преимущественное применение в современных технологиях, подобных FDDI, Fast Ethernet, Gigabit Ethernet, ISDN и т. п., потенциальных избыточных и скрэмблированных кодов вместо манчестерского и биполярного импульсного кода.



N — скорость передачи данных, бит/с;
A — амплитуда сигнала

Рис. 9.9. Спектры потенциальных и импульсных кодов

Компрессия данных

Компрессия, или **сжатие**, данных применяется для сокращения времени их передачи. Так как на компрессию данных передающая сторона тратит дополнительное время, к которому нужно еще прибавить аналогичные затраты времени на декомпрессию этих данных принимающей стороной, то выгоды от сокращения времени на передачу сжатых данных обычно бывают заметны только на низкоскоростных каналах. Соответствующий порог скорости для современной аппаратуры составляет около 64 Кбит/с. Многие программные и аппаратные средства сети способны выполнять *динамическую компрессию* данных в отличие от *статической*, когда данные сначала сжимаются (например, с помощью популярных архиваторов типа WinZip), а уже затем отсылаются в сеть.

На практике может использоваться ряд алгоритмов компрессии, каждый из которых применим к определенному типу данных. Некоторые модемы (называемые интеллектуальными) предлагают **адаптивную компрессию**, при которой в зависимости от передаваемых данных выбирается определенный алгоритм компрессии. Рассмотрим некоторые из общих алгоритмов компрессии данных.

Когда данные состоят только из чисел, значительную экономию можно получить путем уменьшения количества используемых на цифру битов с 7 до 4, просто заменяя десятичные цифры кода ASCII двоичными. Просмотр таблицы кодов ASCII показывает, что старшие три бита всех кодов десятичных цифр содержат комбинацию 011. Если все данные в кадре информации состоят из десятичных цифр, то, поместив в заголовок кадра соответствующий управляющий символ, можно существенно сократить длину кадра. Этот метод носит название **десятичной упаковки**.

Альтернативой десятичной упаковке при передаче числовых данных с небольшими отклонениями между последовательными цифрами является передача только этих откло-

нений вместе с известным опорным значением. Такой метод называется **относительным кодированием** и используется, в частности, при цифровом кодировании голоса с помощью кода ADPCM, когда в каждом такте передается только разница между соседними замерами голоса.

Часто передаваемые данные содержат большое количество повторяющихся байтов. Например, при передаче черно-белого изображения черные поверхности будут порождать большое количество нулевых значений, а максимально освещенные участки изображения — большое количество байтов, состоящих из всех единиц. Передатчик сканирует последовательность передаваемых байтов и если обнаруживает последовательность из трех или более одинаковых байтов, заменяет ее специальной трехбайтовой последовательностью, в которой указывает значение байта, количество его повторений, а также отмечает начало этой последовательности специальным управляющим символом. Этот метод носит название **символьного подавления**.

Метод кодирования с помощью **кодов переменной длины** опирается на тот факт, что не все символы в передаваемом кадре встречаются с одинаковой частотой. Поэтому во многих схемах кодирования коды часто встречающихся символов заменяют кодами меньшей длины, а редко встречающихся — кодами большей длины. Такое кодирование называется также **статистическим кодированием**. Из-за того что символы имеют разную длину, для передачи кадра возможна только бит-ориентированная передача. При статистическом кодировании коды выбираются таким образом, чтобы при анализе последовательности битов можно было бы однозначно определить соответствие определенной порции битов тому или иному символу или же запрещенной комбинации битов. Если данная последовательность битов представляет собой запрещенную комбинацию, то необходимо к ней добавить еще один бит и повторить анализ. Например, если при неравномерном кодировании для наиболее часто встречающегося символа «Р» выбран код 1, состоящий из одного бита, то значение 0 однобитного кода будет запрещенным. Иначе мы сможем закодировать только два символа. Для другого часто встречающегося символа «О» можно использовать код 01, а код 00 оставить как запрещенный. Тогда для символа «А» можно выбрать код 001, для символа «П» — код 0001 и т. п.

Неравномерное кодирование наиболее эффективно, когда неравномерность распределения частот передаваемых символов велика, как при передаче длинных текстовых строк. Напротив, при передаче двоичных данных, например кодов программ, оно малоэффективно, так как 8-битные коды при этом распределены почти равномерно.

Одним из наиболее распространенных алгоритмов, на основе которых строятся неравномерные коды, является **алгоритм Хафмана**, позволяющий строить коды автоматически на основании известных частот появления символов. Существуют адаптивные модификации метода Хафмана, которые позволяют строить дерево кодов «на ходу», по мере поступления данных от источника.

Многие модели коммуникационного оборудования, такие как модемы, мосты, коммутаторы и маршрутизаторы, поддерживают протоколы динамической компрессии, позволяющие сократить объем передаваемой информации в 4, а иногда и в 8 раз. В таких случаях говорят, что протокол обеспечивает коэффициент сжатия 1:4 или 1:8. Существуют стандартные протоколы компрессии, например V.42bis, а также большое количество нестандартных фирменных протоколов. Реальный коэффициент компрессии зависит от типа передаваемых данных. Так, графические и текстовые данные обычно сжимаются хорошо, а коды программ — хуже.

Обнаружение и коррекция ошибок

Надежную передачу информации обеспечивают различные методы. В главе 6 были рассмотрены принципы работы протоколов, которые обеспечивают надежность за счет повторной передачи искаженных или потерянных пакетов. Такие протоколы основаны на том, что приемник в состоянии распознать факт искажения информации в принятом кадре. Еще одним, более эффективным подходом, чем повторная передача пакетов, является использование самокорректирующихся кодов, которые позволяют не только обнаруживать, но и исправлять ошибки в принятом кадре.

Методы обнаружения ошибок

Методы обнаружения ошибок основаны на передаче в составе блока данных избыточной служебной информации, по которой можно судить с некоторой степенью вероятности о достоверности принятых данных. В сетях с коммутацией пакетов такой единицей информации может быть PDU любого уровня, для определенности будем считать, что мы контролируем кадры.

Избыточную служебную информацию принято называть **контрольной суммой**, или **контрольной последовательностью кадра** (Frame Check Sequence, FCS). Контрольная сумма вычисляется как функция от основной информации, причем *не обязательно путем суммирования*. Принимающая сторона повторно вычисляет контрольную сумму кадра по известному алгоритму и в случае ее совпадения с контрольной суммой, вычисленной передающей стороной, делает вывод о том, что данные были переданы через сеть корректно. Рассмотрим несколько распространенных алгоритмов вычисления контрольной суммы, отличающихся вычислительной сложностью и способностью обнаруживать ошибки в данных.

Контроль по паритету представляет собой наиболее простой метод контроля данных. В то же время это наименее мощный алгоритм контроля, так как с его помощью можно обнаруживать только одиночные ошибки в проверяемых данных. Метод заключается в суммировании по модулю 2 всех битов контролируемой информации. Нетрудно заметить, что для информации, состоящей из нечетного числа единиц, контрольная сумма всегда равна 1, а при четном числе единиц — 0. Например, для данных 100101011 результатом контрольного суммирования будет значение 1. Результат суммирования также представляет собой один дополнительный бит данных, который пересылается вместе с контролируемой информацией. При искажении в процессе пересылки любого одного бита исходных данных (или контрольного разряда) результат суммирования будет отличаться от принятого контрольного разряда, что говорит об ошибке. Однако двойная ошибка, например 110101010, будет неверно принята за корректные данные. Поэтому контроль по паритету применяется к небольшим порциям данных, как правило, к каждому байту, что дает коэффициент избыточности для этого метода 1/8. Метод редко используется в компьютерных сетях из-за значительной избыточности и невысоких диагностических возможностей.

Вертикальный и горизонтальный контроль по паритету представляет собой модификацию описанного метода. Его отличие состоит в том, что исходные данные рассматриваются в виде матрицы, строки которой составляют байты данных. Контрольный разряд подсчитывается отдельно для каждой строки и для каждого столбца матрицы. Этот метод позволяет обнаруживать большую часть двойных ошибок, однако он обладает еще большей избыточностью. На практике этот метод сейчас также почти не применяется при передаче информации по сети.

Циклический избыточный контроль (Cyclic Redundancy Check, CRC) является в настоящее время наиболее популярным методом контроля в вычислительных сетях (и не только в сетях, например, этот метод широко применяется при записи данных на гибкие и жесткие диски). Метод основан на представлении исходных данных в виде одного многоразрядного двоичного числа. Например, кадр стандарта Ethernet, состоящий из 1024 байт, рассматривается как одно число, состоящее из 8192 бит. Контрольной информацией считается остаток от деления этого числа на известный делитель *R*. Обычно в качестве делителя выбирается семнадцати- или тридцатитрехразрядное число, чтобы остаток от деления имел длину 16 разрядов (2 байт) или 32 разряда (4 байт). При получении кадра данных снова вычисляется остаток от деления на тот же делитель *R*, но при этом к данным кадра добавляется и содержащаяся в нем контрольная сумма. Если остаток от деления на *R* равен нулю, то делается вывод об отсутствии ошибок в полученном кадре, в противном случае кадр считается искаженным.

Этот метод обладает более высокой вычислительной сложностью, но его диагностические возможности гораздо выше, чем у методов контроля по паритету. Метод CRC позволяет обнаруживать все одиночные ошибки, двойные ошибки и ошибки в нечетном числе битов. Метод обладает также невысокой степенью избыточности. Например, для кадра Ethernet размером 1024 байт контрольная информация длиной 4 байт составляет только 0,4 %.

Методы коррекции ошибок

Техника кодирования, которая позволяет приемнику не только понять, что присланные данные содержат ошибки, но и исправить их, называется **прямой коррекцией ошибок** (Forward Error Correction, FEC). Коды, которые обеспечивают прямую коррекцию ошибок, требуют введения большей избыточности в передаваемые данные, чем коды, только обнаруживающие ошибки.

При применении любого избыточного кода не все комбинации кодов являются разрешенными. Например, контроль по паритету делает разрешенными только половину кодов. Если мы контролируем три информационных бита, то разрешенными 4-битными кодами с дополнением до нечетного количества единиц будут:

000 1, 001 0, 010 0, 011 1, 100 0, 101 1, 110 1, 111 0

То есть всего 8 кодов из 16 возможных.

Для того чтобы оценить количество дополнительных битов, требуемых для исправления ошибок, нужно знать так называемое расстояние Хемминга между разрешенными комбинациями кода. **Расстоянием Хемминга** называется минимальное число битовых разрядов, в которых отличается любая пара разрешенных кодов. Для схем контроля по паритету расстояние Хемминга равно 2.

Можно доказать, что если мы сконструировали избыточный код с расстоянием Хемминга, равным *n*, то такой код будет в состоянии распознавать $(n-1)$-кратные ошибки и исправлять $(n-1)/2$-кратные ошибки. Так как коды с контролем по паритету имеют расстояние Хемминга, равное 2, то они могут только обнаруживать однократные ошибки и не могут исправлять ошибки.

Коды Хемминга эффективно обнаруживают и исправляют изолированные ошибки, то есть отдельные искаженные биты, которые разделены большим количеством корректных

битов. Однако при появлении длинной последовательности искаженных битов (пульсации ошибок) коды Хемминга не работают.

Пульсации ошибок характерны для *беспроводных каналов*, в которых применяют **сверточные коды**. Поскольку для распознавания наиболее вероятного корректного кода в этом методе задействуется решетчатая диаграмма, то такие коды еще называют **решетчатыми**. Эти коды используются не только в беспроводных каналах, но и в модемах.

Методы прямой коррекции ошибок особенно эффективны для технологий физического уровня, которые не поддерживают сложные процедуры повторной передачи данных в случае их искажения. Примерами таких технологий являются технологии SDH и OTN, рассматриваемые в главе 11.

Мультиплексирование и коммутация

Методы кодирования и коррекции ошибок позволяют создать в некоторой среде, например в медных проводах кабеля, линию связи. Однако для эффективного соединения пользователей сети этого недостаточно. Нужно образовать в этой линии отдельные каналы передачи данных, служащие для коммутации информационных потоков пользователей. Для создания пользовательского канала коммутаторы первичных сетей должны поддерживать какую-либо технику мультиплексирования и коммутации. Методы коммутации тесно связаны с выбранным методом мультиплексирования, поэтому здесь они изучаются совместно.

В настоящее время для мультиплексирования абонентских каналов используются:

- частотное мультиплексирование (Frequency Division Multiplexing, FDM);
- волновое мультиплексирование (Wave Division Multiplexing, WDM);
- временное мультиплексирование (Time Division Multiplexing, TDM);
- множественный доступ с кодовым разделением (Code Division Multiple Access, CDMA).

Метод TDM используется при коммутации как каналов, так и пакетов. Методы FDM, WDM и CDMA пригодны исключительно для коммутации каналов. Метод CDMA применяется только в технике расширенного спектра и рассматривается в следующей главе, посвященной беспроводной передаче.

Коммутация каналов на основе методов FDM и WDM

Техника **частотного мультиплексирования** (FDM) была разработана для телефонных сетей, но применяется она и для других видов сетей, например первичных сетей (микроволновые каналы) или сетей кабельного телевидения.

Основная идея этого метода состоит в выделении каждому соединению собственного диапазона частот в общей полосе пропускания линии связи.

На основе этого диапазона создается **канал**. Данные, передаваемые в канале, модулируются с помощью одного из описанных ранее методов с использованием несущей частоты, при-

надлежащей диапазону канала. Мультиплексирование выполняется с помощью смесителя частот, а демультплексирование — с помощью узкополосного фильтра, ширина которого равна ширине диапазона канала.

Рассмотрим особенности этого вида мультиплексирования на примере телефонной сети. На входы FDM-коммутатора поступают исходные сигналы от абонентов телефонной сети. Коммутатор переносит частоту каждого канала в выделенный каналу диапазон за счет модуляции определенной несущей частоты. Чтобы низкочастотные составляющие сигналов разных каналов не смешивались между собой, полосы делают шириной в 4 кГц, а не в 3,1 кГц, оставляя между ними страховочный промежуток в 900 Гц (рис. 9.10). В линии связи между двумя FDM-коммутаторами одновременно передаются сигналы всех абонентских каналов, но каждый из них занимает *свою* полосу частот. Такой канал называют **уплотненным**.

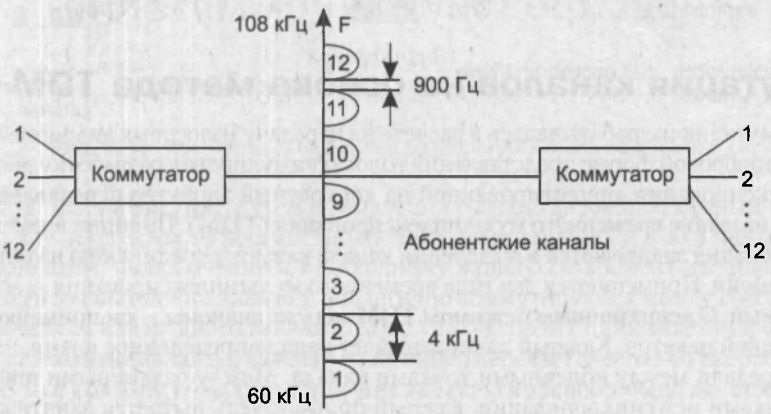


Рис. 9.10. FDM-коммутация

Выходной FDM-коммутатор выделяет модулированные сигналы каждой несущей частоты и передает их на соответствующий выходной канал, к которому непосредственно подключен абонентский телефон.

FDM-коммутаторы могут выполнять как динамическую, так и постоянную коммутацию. При *динамической коммутации* один абонент инициирует соединение с другим абонентом, посылая в сеть номер вызываемого абонента. Коммутатор динамически выделяет данному абоненту одну из свободных полос своего уплотненного канала. При постоянной коммутации за абонентом полоса в 4 кГц закрепляется на длительный срок путем настройки коммутатора по отдельному входу, недоступному пользователям.

Принцип коммутации на основе разделения частот остается неизменным и в сетях другого вида, меняются только границы полос, выделяемых отдельному абонентскому каналу, а также количество низкоскоростных каналов в высокоскоростном канале.

В методе **волнового мультиплексирования** (WDM) используется тот же принцип частотного разделения каналов, но только в другой области электромагнитного спектра. Информационным сигналом является не электрический ток и не радиоволны, а свет. Для организации WDM-каналов в волоконно-оптическом кабеле задействуют волны инфракрасного диапазона длиной от 850 до 1565 нм, что соответствует частотам от 196 до 350 ТГц.

В магистральном канале обычно мультиплексируется несколько спектральных каналов — до 16, 32, 40, 80 или 160, причем, начиная с 16 каналов, такая техника мультиплексирования называется **уплотненным волновым мультиплексированием** (Dense Wave Division Multiplexing, DWDM). Внутри такого спектрального канала данные могут кодироваться как дискретным способом, так и аналоговым. По сути WDM и DWDM — это реализации идеи частотного аналогового мультиплексирования, но в другой форме. Отличие сетей WDM/DWDM от сетей FDM заключается в предельных скоростях передачи информации. Если сети FDM обычно обеспечивают на магистральных каналах одновременную передачу до 600 разговоров, что соответствует суммарной скорости в 36 Мбит/с (для сравнения с цифровыми каналами скорость пересчитана из расчета 64 Кбит/с на один разговор), то сети DWDM обеспечивают общую пропускную способность до сотен гигабитов и даже нескольких терабитов в секунду.

Более подробно технология DWDM рассматривается в главе 11.

Коммутация каналов на основе метода TDM

FDM-коммутация разрабатывалась в расчете на передачу голосовых аналоговых сигналов. Переход к цифровой форме представления голоса стимулировал разработку новой техники мультиплексирования, ориентированной на дискретный характер передаваемых данных и носящей название **временного мультиплексирования** (TDM). Принцип временного мультиплексирования заключается в выделении канала каждому соединению на определенный период времени. Применяются два типа временного мультиплексирования — асинхронный и синхронный. С **асинхронным режимом TDM** мы уже знакомы — он применяется в сетях с коммутацией пакетов. Каждый пакет занимает канал определенное время, необходимое для его передачи между конечными точками канала. Между различными информационными потоками нет синхронизации, каждый пользователь пытается занять канал тогда, когда у него возникает потребность в передаче информации.

Рассмотрим теперь **синхронный режим TDM**[1]. В этом случае доступ всех информационных потоков к каналу синхронизируется таким образом, чтобы каждый информационный поток периодически получал канал в свое распоряжение на фиксированный промежуток времени.

Рисунок 9.11 поясняет принцип коммутации каналов на основе техники TDM при передаче голоса.

Аппаратура TDM-сетей — мультиплексоры, коммутаторы, демультиплексоры — работает в режиме разделения времени, поочередно обслуживая в течение цикла своей работы все абонентские каналы. Цикл равен 125 мкс, что соответствует периоду следования замеров голоса в цифровом абонентском канале. Это значит, что мультиплексор или коммутатор успевает вовремя обслужить любой абонентский канал и передать его очередной замер далее по сети. Каждому соединению выделяется один квант времени цикла работы аппаратуры, называемый также **тайм-слотом**. Длительность тайм-слота зависит от числа абонентских каналов, обслуживаемых мультиплексором или коммутатором.

---

[1] Когда аббревиатура TDM используется без уточнения режима работы, то она всегда обозначает синхронный режим TDM.

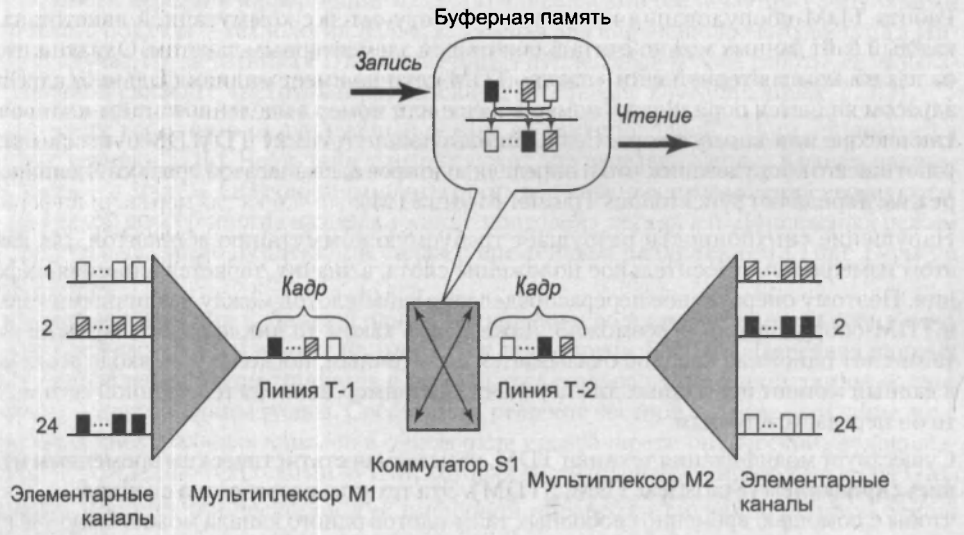


Рис. 9.11. Коммутация на основе разделения канала во времени

В сети, показанной на рисунке, путем коммутации создано 24 канала, каждый из которых связывает пару абонентов. В частности, абонент, подключенный к входному каналу 1, связан с абонентом, подключенным к выходному каналу 24, абонент входного канала 2 связан с абонентом выходного канала 1, аналогично коммутируются между собой абоненты входного канала 24 и выходного канала 2. Мультиплексор *M*1 принимает информацию от абонентов по входным каналам, каждый из которых передает данные со скоростью 1 байт каждые 125 мкс (64 Кбит/с). В каждом цикле мультиплексор выполняет следующие действия:

1. Прием от каждого канала очередного байта данных.
2. Составление из принятых байтов кадра.
3. Передача кадра на выходной канал с битовой скоростью, равной 24 × 64 Кбит/с, что примерно составляет 1,5 Мбит/с.

Порядок следования байта в кадре соответствует номеру входного канала, от которого этот байт получен. Коммутатор *S*1 принимает кадр по скоростному каналу от мультиплексора и записывает каждый байт из него в отдельную ячейку своей буферной памяти, причем в том порядке, в котором байты были упакованы в уплотненный кадр. Для выполнения коммутации байты извлекаются из буферной памяти не в порядке поступления, а в том порядке, который соответствует поддерживаемым в сети соединениям абонентов. В рассматриваемом примере коммутатор *S*1 коммутирует входные каналы 1, 2 и 24 с выходными каналами 24, 2 и 1 соответственно. Для выполнения этой операции первым из буферной памяти должен быть извлечен байт 2, вторым — байт 24, а последним — байт 1. «Перемешивая» нужным образом байты в кадре, коммутатор обеспечивает требуемое соединение абонентов в сети.

Мультиплексор *M*2 решает обратную задачу — он разбирает байты кадра и распределяет их по своим нескольким выходным каналам, при этом он также считает, что порядковый номер байта в кадре соответствует номеру выходного канала.

Работа TDM-оборудования напоминает работу сетей с коммутацией пакетов, так как каждый байт данных можно считать некоторым элементарным пакетом. Однако в отличие от пакета компьютерной сети «пакет» TDM-сети не имеет индивидуального адреса. Его адресом является порядковый номер в кадре или номер выделенного тайм-слота в мультиплексоре или коммутаторе. Сети, использующие технику TDM, требуют синхронной работы всего оборудования, что и определило второе название этой техники — **синхронный режим передачи** (Synchronous Transfer Mode, STM).

Нарушение синхронности разрушает требуемую коммутацию абонентов, так как при этом изменяется относительное положение слота, а значит, теряется адресная информация. Поэтому оперативное перераспределение тайм-слотов между различными каналами в TDM-оборудовании невозможно. Даже если в каком-то цикле работы мультиплексора тайм-слот одного из каналов оказывается избыточным, поскольку на входе этого канала в данный момент нет данных для передачи (например, абонент телефонной сети молчит), то он передается пустым.

Существует модификация техники TDM, называемая **статистическим временным мультиплексированием** (Statistical TDM, STDM). Эта техника разработана специально для того, чтобы с помощью временно свободных тайм-слотов одного канала можно было увеличить пропускную способность остальных. Для решения этой задачи каждый байт данных дополняется полем адреса небольшой длины, например в 4 или 5 бит, что позволяет мультиплексировать 16 или 32 канала. Фактически STDM представляет собой уже технику коммутации пакетов, но только с очень упрощенной адресацией и узкой областью применения. Техника STDM не стала популярной и используется в основном в нестандартном оборудовании подключения терминалов к мэйнфреймам. Развитием идей статистического мультиплексирования стала **технология асинхронного режима передачи** (Asynchronous Transfer Mode, ATM), которая относится уже к коммутации пакетов.

TDM-сети могут поддерживать режим динамической или постоянной коммутации, а иногда и оба эти режима. Основным режимом цифровых телефонных сетей, работающих на основе технологии TDM, является динамическая коммутация, но они поддерживают также и постоянную коммутацию, предоставляя своим абонентам выделенную линию.

Дуплексный режим работы канала

Дуплексный режим — это наиболее универсальный и производительный режим работы канала. Самым простым вариантом организации дуплексного режима является использование двух независимых линий связи (двух пар проводников или двух оптических волокон) в кабеле, каждая из которых работает в симплексном режиме, то есть передает данные в одном направлении. Именно такая идея лежит в основе реализации дуплексного режима работы многих сетевых технологий, например Fast Ethernet или ATM.

Иногда такое простое решение оказывается недоступным или неэффективным, например, когда прокладка второй линии связи ведет к большим затратам. Так, при обмене данными с помощью модемов через телефонную сеть у пользователя имеется только одна линия связи с телефонной станцией — двухпроводная. В таких случаях дуплексный режим работы организуется на основе разделения линии связи на два логических канала с помощью техники FDM или TDM.

При использовании техники FDM для организации дуплексного канала диапазон частот делится на две части. Деление может быть симметричным и асимметричным, в последнем

случае скорости передачи информации в каждом направлении различаются (популярный пример такого подхода — технология ADSL, служащая для широкополосного доступа в Интернет). В случае, когда техника FDM обеспечивает дуплексный режим работы, ее называют **дуплексной связью с частотным разделением** (Frequency Division Duplex, FDD).

При цифровом кодировании дуплексный режим на двухпроводной линии организуется с помощью техники TDM. Часть тайм-слотов служит для передачи данных в одном направлении, часть — в другом. Обычно тайм-слоты противоположных направлений чередуются, из-за чего такой способ иногда называют «пинг-понговой» передачей. Дуплексный режим TDM получил название **дуплексной связи с временны́м разделением** (Time Division Duplex, TDD).

В волоконно-оптических кабелях с одним оптическим волокном для организации дуплексного режима работы может применяться технология DWDM. Передача данных в одном направлении осуществляется с помощью светового пучка одной длины волны, в обратном — другой длины волны. Собственно, решение частной задачи — создание двух независимых спектральных каналов в одном окне прозрачности оптического волокна — и привело к рождению технологии WDM, которая затем трансформировалась в DWDM.

Появление мощных процессоров для цифровой обработки сигналов (Digital Signal Processor, DSP), способных выполнять сложные алгоритмы обработки сигналов в реальном времени, сделало возможным еще один вариант дуплексной работы. Два передатчика работают одновременно навстречу друг другу, создавая в канале суммарный аддитивный сигнал. Так как каждый передатчик знает спектр собственного сигнала, то он вычитает его из суммарного сигнала, получая в результате сигнал, посылаемый другим передатчиком.

Выводы

Для представления дискретной информации применяются сигналы двух типов: прямоугольные импульсы и синусоидальные волны. В первом случае используют термин «кодирование», во втором — «модуляция».

При модуляции дискретной информации единицы и нули кодируются изменением амплитуды, частоты или фазы синусоидального сигнала.

Аналоговая информация может передаваться по линиям связи в цифровой форме. Это повышает качество передачи, так как позволяет применять эффективные методы обнаружения и исправления ошибок, недоступные для систем аналоговой передачи. Для качественной передачи голоса в цифровой форме используется частота оцифровывания в 8 кГц, когда каждое значение амплитуды голоса представляется 8-битным числом. Это определяет скорость голосового канала в 64 Кбит/с.

При выборе способа кодирования нужно одновременно стремиться к достижению нескольких целей: минимизировать возможную ширину спектра результирующего сигнала, обеспечивать синхронизацию между передатчиком и приемником, обеспечивать устойчивость к шумам, обнаруживать и по возможности исправлять битовые ошибки, минимизировать мощность передатчика.

Спектр сигнала является одной из наиболее важных характеристик способа кодирования. Более узкий спектр сигналов позволяет добиваться более высокой скорости передачи данных при фиксированной полосе пропускания среды.

Код должен обладать свойством самосинхронизации, то есть сигналы кода должны содержать признаки, по которым приемник может определить, в какой момент времени нужно осуществлять распознавание очередного бита.

При дискретном кодировании двоичная информация представляется различными уровнями постоянного потенциала или полярностью импульса.

Наиболее простым потенциальным кодом является код без возвращения к нулю (NRZ), однако он не является самосинхронизирующимся.

Для улучшения свойств потенциального кода NRZ используются методы, основанные на введении избыточных битов в исходные данные и на скрэмблировании исходных данных.

Коды Хэмминга и сверточные коды позволяют не только обнаруживать, но и исправлять многократные ошибки. Эти коды наиболее часто используются для прямой коррекции ошибок (FEC).

Для повышения полезной скорости передачи данных в сетях применяется динамическая компрессия данных на основе различных алгоритмов. Коэффициент сжатия зависит от типа данных и применяемого алгоритма и может колебаться в пределах от 1:2 до 1:8.

Для образования нескольких каналов в линии связи используются различные методы мультиплексирования, включая частотное (FDM), временно́е (TDM) и волновое (WDM) мультиплексирование, а также множественный доступ с кодовым разделением (CDMA). Техника коммутации пакетов сочетается только с методом TDM, а техника коммутации каналов позволяет использовать любой тип мультиплексирования.

Вопросы и задания

1. Сколько частот используется в методе модуляции BFSK?
2. Какие параметры синусоиды изменяются в методе QAM? Варианты ответов:
   а) амплитуда и фаза;
   б) амплитуда и частота;
   в) частота и фаза.
3. Для какой цели в решетчатых кодах добавляется 5-й бит?
4. Сколько битов передает один символ кода, имеющий 10 состояний?
5. Поясните, из каких соображений выбрана частота дискретизации 8 кГц в методе квантования PCM?
6. При каком методе кодирования/модуляции спектр сигнала симметричен относительно основной гармоники? Варианты ответов:
   а) потенциальное кодирование;
   б) амплитудная модуляция;
   в) фазовая модуляция.
7. Какой способ применяется для улучшения самосинхронизации кода B8ZS?
8. Чем логическое кодирование отличается от физического?
9. Каким образом можно повысить скорость передачи данных по кабельной линии связи? Варианты ответов:
   а) сузить спектр сигнала за счет применения другого метода кодирования/модуляции и повысить тактовую частоту сигнала;
   б) применить кабель с более широкой полосой пропускания и повысить тактовую частоту сигнала;
   в) увеличить спектр сигнала за счет применения другого метода кодирования и повысить тактовую частоту сигнала.

10. По каким причинам код NRZ не применяется в телекоммуникационных сетях?
11. Какими способами можно улучшить свойство самосинхронизации кода NRZI? Варианты ответов:

    а) скремблировать данные;

    б) использовать логическое кодирование, исключающее появление длинных последовательностей единиц;

    в) использовать логическое кодирование, исключающее появление длинных последовательностей нулей.
12. Какое значение бита кодируется в манчестерском коде перепадом от низкого уровня сигнала к высокому? Варианты ответов:

    а) единица; б) нуль.
13. Какой принцип лежит в основе методов обнаружения и коррекции ошибок? Варианты ответов:

    а) самосинхронизация;

    б) избыточность;

    в) максимизация отношения мощности сигнала к мощности помех.
14. Каково расстояние Хемминга в схемах контроля по паритету?
15. Предложите избыточный код с расстоянием Хемминга, равным 3.
16. Какой режим временного мультиплексирования используется в сетях с коммутацией пакетов?
17. Найдите первые две гармоники спектра NRZ-сигнала при передаче последовательности 110011001100..., если тактовая частота передатчика равна 100 МГц.
18. Какие из 16-ти кодов 3B/4B вы выберете для передачи пользовательской информации?
19. Могут ли данные надежно передаваться по каналу с полосой пропускания от 2,1 до 2,101 ГГц, если для их передачи используются несущая частота 2,1005 ГГц, амплитудная манипуляция с двумя значениями амплитуды и тактовая частота 5 МГц?
20. Предложите коды неравной длины для каждого из символов A, B, C, D, F и O, если нужно передать сообщение BDDACAAFOOOAOOOO. Будет ли достигнута компрессия данных по сравнению с использованием:

    а) традиционных кодов ASCI;

    б) кодов равной длины, учитывающих наличие только данных символов.
21. Во сколько раз увеличится ширина спектра кода NRZ при увеличении тактовой частоты передатчика в 2 раза?

ГЛАВА 10 Беспроводная передача данных

Беспроводная связь стала использоваться для общения между людьми ненамного позже, чем проводная. Уже в 90-х годах XIX века были проведены первые эксперименты по передаче телеграфных сообщений с помощью радиосигналов, а в 20-е годы XX века началось применение радио для передачи голоса.

Сегодня существует большое число беспроводных телекоммуникационных систем, из которых наиболее распространенными являются системы широковещания, такие как радио или телевидение, а также мобильная телефонная связь. Кроме того, беспроводные системы широко используются как транспортное средство для передачи компьютерных данных. Для создания протяженных линий связи применяются радиорелейные и спутниковые системы, существуют также беспроводные системы доступа к сетям операторов связи и беспроводные локальные сети. В современных беспроводных системах, так же как и в проводных, все больше информации передается в цифровом виде.

Беспроводная среда, для которой сегодня в основном используется микроволновый диапазон, отличается высоким уровнем помех, которые создают внешние источники излучения, а также многократно отраженные от стен и других преград полезные сигналы. Поэтому в беспроводных системах связи применяют различные средства, направленные на снижение влияния помех. В арсенал таких средств входят уже рассмотренные нами коды прямой коррекции ошибок и протоколы с подтверждением доставки информации. Эффективным средством борьбы с помехами является техника расширенного спектра, разработанная специально для беспроводных систем.

В этой главе приводятся базовые сведения об элементах, принципах работы и методах кодирования беспроводных систем, которые используются для построения двухточечных и многоточечных линий связи.

Беспроводная среда передачи

Преимущества беспроводных коммуникаций

Возможность передавать информацию без проводов, привязывающих (в буквальном смысле этого слова) абонентов к определенной точке пространства, всегда была очень привлекательной. И как только технические возможности становились достаточными для того, чтобы новый вид беспроводных услуг приобрел две необходимые составляющие успеха — удобство использования и низкую стоимость — успех ему был гарантирован.

Последнее тому доказательство — **мобильная телефония**. Первый мобильный телефон был изобретен еще в 1910 году Ларсом Магнусом Эрикссоном (Lars Magnus Ericsson). Этот телефон предназначался для автомобиля и был беспроводным только во время движения. Однако в движении им нельзя было пользоваться, для разговора нужно было остановиться, выйти из автомобиля и с помощью длинных жердей присоединить телефон к придорожным телефонным проводам (рис. 10.1). Понятно, что определенные неудобства и ограниченная мобильность воспрепятствовали коммерческому успеху этого вида телефонии.

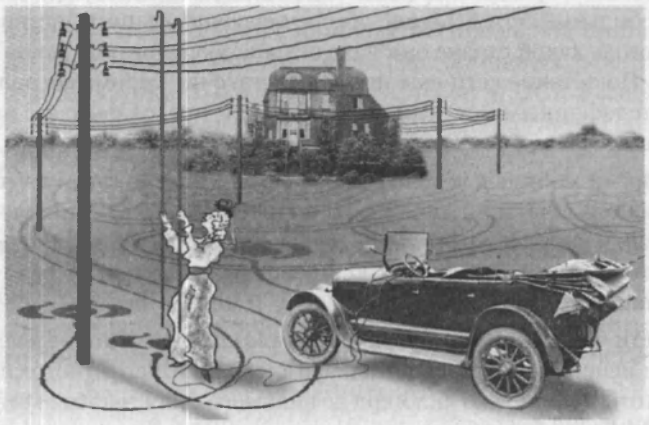


Рис. 10.1. Первый мобильный телефон

Прошло много лет, прежде чем технологии радиодоступа достигли определенной степени зрелости и в конце 70-х обеспечили производство сравнительно компактных и недорогих радиотелефонов. С этого времени начался бум мобильной телефонии, который продолжается до настоящего времени.

Беспроводная связь не обязательно означает мобильность. Существует так называемая **фиксированная беспроводная связь**, когда взаимодействующие узлы постоянно располагаются в пределах небольшой территории, например в определенном здании. Фиксированная беспроводная связь применяется вместо проводной, когда по какой-то причине невозможно или невыгодно использовать кабельные линии связи. Причины могут быть разными. Например, малонаселенная или труднодоступная местность — болотистые районы и джунгли Бразилии, пустыни, крайний Север или Антарктида еще не скоро дождутся своих кабельных систем. Другой пример — здания, имеющие историческую ценность, стены которых непозволительно подвергать испытанию прокладкой кабеля.

Еще один часто встречающийся случай использования фиксированной беспроводной связи — получение альтернативным оператором связи доступа к абонентам, дома которых уже подключены к точкам присутствия существующего уполномоченного оператора связи проводными линиями доступа. Наконец, организация временной связи, например, при проведении конференции в здании, в котором отсутствует проводной канал, имеющий скорость, достаточную для качественного обслуживания многочисленных участников конференции.

Беспроводная связь используется для передачи данных уже достаточно давно. До недавнего времени большая часть применений беспроводной связи в компьютерных сетях была связана с ее фиксированным вариантом. Не всегда архитекторы и пользователи компьютерной сети знают о том, что на каком-то участке пути данные передаются не по проводам, а распространяются в виде электромагнитных колебаний через атмосферу или космическое пространство. Это может происходить в том случае, когда компьютерная сеть арендует линию связи у оператора первичной сети, и отдельный канал такой линии является спутниковым или наземным СВЧ-каналом.

Начиная с середины 90-х годов достигла необходимой зрелости и технология **мобильных компьютерных сетей**. С появлением стандарта IEEE 802.11 в 1997 году появилась возможность строить мобильные сети Ethernet, обеспечивающие взаимодействие пользователей независимо от того, в какой стране они находятся и оборудование какого производителя они применяют. Пока такие сети еще играют достаточно скромную роль по сравнению с мобильными телефонными сетями, но аналитики предсказывают их быстрый рост в ближайшие годы.

Развитие технологии мобильных телефонных сетей привело к тому, что эти сети стали очень широко использоваться для доступа в Интернет. Третье поколение мобильных телефонных сетей, известное как сети 3G, обеспечивает передачу данных со скоростью 1,5–2 Мбит/с, что сравнимо по скорости с проводным доступом через телефонные абонентские окончания.

Беспроводные сети часто связывают с *радиосигналами*, однако это не всегда верно. В беспроводной связи используется широкий диапазон электромагнитного спектра, от радиоволн низкой частоты в несколько килогерц до видимого света, частота которого составляет примерно 8 × 1014 Гц.

Беспроводная линия связи

Беспроводная линия связи строится в соответствии с достаточно простой схемой (рис. 10.2).

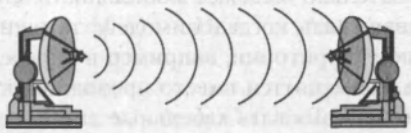


Рис. 10.2. Беспроводная линия связи

Каждый узел оснащается антенной, которая одновременно является передатчиком и приемником электромагнитных волн. Электромагнитные волны распространяются в ат-

мосфере или вакууме со скоростью 3 × 108 м/с во всех направлениях или же в пределах определенного сектора.

Направленность или ненаправленность распространения зависит от типа антенны. На рис. 10.2 показана **параболическая антенна**, которая является *направленной*. Другой тип антенн — **изотропная антенна**, представляющая собой вертикальный проводник длиной в четверть волны излучения. Изотропные антенны являются *ненаправленными*, они широко используются в автомобилях и портативных устройствах. Распространение излучения во всех направлениях можно также обеспечить несколькими направленными антенными.

Так как при ненаправленном распространении электромагнитные волны заполняют все пространство (в пределах определенного радиуса, определяемого затуханием мощности сигнала), то это пространство может служить *разделяемой средой*. Разделение среды передачи порождает те же проблемы, что и в локальных сетях, однако здесь они усугубляются тем, что пространство в отличие от кабеля является общедоступным, а не принадлежит одной организации.

Кроме того, проводная среда строго определяет направление распространения сигнала в пространстве, а *беспроводная среда является ненаправленной*.

Для передачи дискретной информации с помощью беспроводной линии связи необходимо модулировать электромагнитные колебания передатчика в соответствии с потоком передаваемых битов. Эту функцию осуществляет устройство DCE, располагаемое между антенной и устройством DTE, которым может быть компьютер, коммутатор или маршрутизатор компьютерной сети.

Диапазоны электромагнитного спектра

Характеристики беспроводной линии связи — расстояние между узлами, территория охвата, скорость передачи информации и т. п. — во многом зависят от частоты используемого электромагнитного спектра (частота f и длина волны λ связаны соотношением $c = f \times \lambda$).

На рис. 10.3 показаны диапазоны электромагнитного спектра. Обобщая можно сказать, что они и соответствующие им беспроводные системы передачи информации делятся на четыре группы.

- Диапазон до 300 ГГц имеет общее стандартное название — **радиодиапазон**. Союз ITU разделил его на несколько поддиапазонов (они показаны на рисунке), начиная от сверхнизких частот (Extremely Low Frequency, ELF) и заканчивая сверхвысокими (Extra High Frequency, EHF). Привычные для нас радиостанции работают в диапазоне от 20 кГц до 300 МГц, и для этих диапазонов существует хотя и не определенное в стандартах, однако часто используемое название **широковещательное радио**. Сюда попадают низкоскоростные системы AM- и FM-диапазонов, предназначенные для передачи данных со скоростями от нескольких десятков до сотен килобитт в секунду. Примером могут служить радиомодемы, которые соединяют два сегмента локальной сети на скоростях 2400, 9600 или 19200 Кбит/с.
- Несколько диапазонов от 300 МГц до 300 ГГц имеют также нестандартное название микроволновых диапазонов. **Микроволновые системы** представляют наиболее широкий класс систем, объединяющий радиорелейные линии связи, спутниковые каналы, беспроводные локальные сети и системы фиксированного беспроводного доступа,

называемые также системами беспроводных абонентских окончаний (Wireless Local Loop, WLL).

- Выше микроволновых диапазонов располагается инфракрасный диапазон. Микроволновые и инфракрасный диапазоны также широко используются для беспроводной передачи информации. Так как инфракрасное излучение не может проникать через стены, то **системы инфракрасных волн** служат для образования небольших сегментов локальных сетей в пределах одного помещения.
- В последние годы видимый свет тоже стал применяться для передачи информации (с помощью лазеров). **Системы видимого света** используются как высокоскоростная альтернатива микроволновым двухточечным каналам для организации доступа на небольших расстояниях.

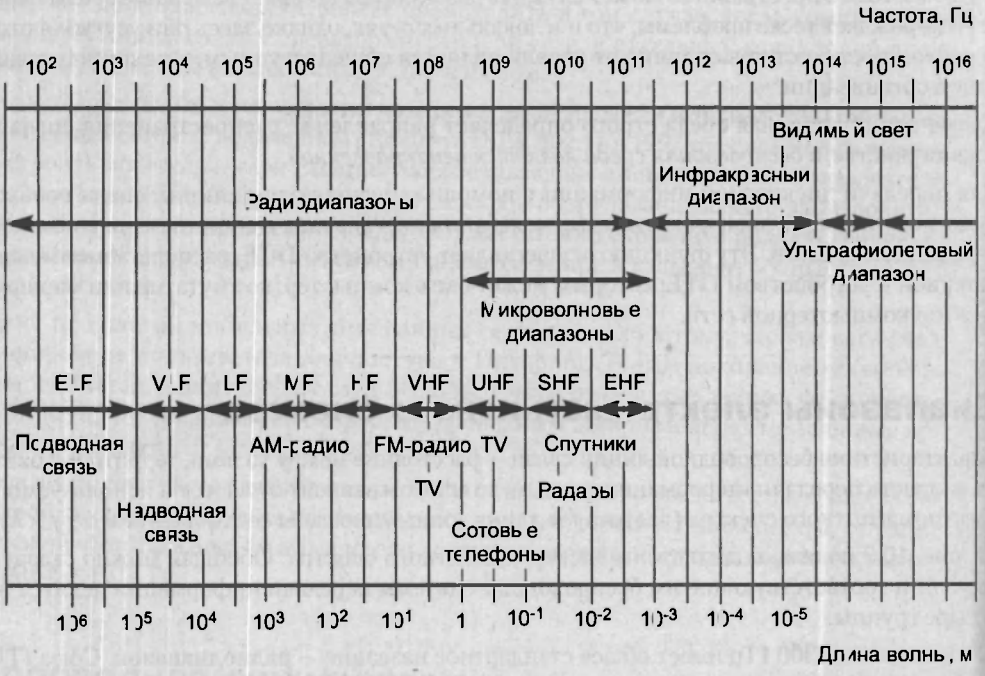


Рис. 10.3. Диапазоны электромагнитного спектра

ПРИМЕЧАНИЕ

Справедливости ради нужно отметить, что свет был, очевидно, первой беспроводной средой передачи информации, так как он использовался в древних цивилизациях (например, в Древней Греции) для эстафетной передачи сигналов между цепочкой наблюдателей, располагавшихся на вершинах холмов.

Распространение электромагнитных волн

Перечислим некоторые общие закономерности распространения электромагнитных волн, связанные с частотой излучения.

- Чем выше несущая частота, тем выше возможная скорость передачи информации.
- Чем выше частота, тем хуже проникает сигнал через препятствия. Низкочастотные радиоволны AM-диапазонов легко проникают в дома, позволяя обходиться комнатной антенной. Более высокочастотный сигнал телевидения требует, как правило, внешней антенны. И наконец, инфракрасный и видимый свет не проходят через стены, ограничивая передачу *прямой видимостью* (Line Of Sight, LOS).
- Чем выше частота, тем быстрее убывает энергия сигнала с расстояниям от источника. При распространении электромагнитных волн в свободном пространстве (без отражений) затухание мощности сигнала пропорционально произведению квадрата расстояния от источника сигнала на квадрат частоты сигнала.
- Низкие частоты (до 2 МГц) распространяются вдоль поверхности земли. Именно поэтому сигналы AM-радио могут передаваться на расстояния в сотни километров.
- Сигналы частот от 2 до 30 МГц отражаются ионосферой земли, поэтому они могут распространяться даже на более значительные расстояния в несколько тысяч километров (при достаточной мощности передатчика).
- Сигналы в диапазоне выше 30 МГц распространяются только по прямой, то есть являются сигналами прямой видимости. При частоте свыше 4 ГГц их подстерегает неприятность — они начинают поглощаться водой, а это означает, что не только дождь, но и туман может стать причиной резкого ухудшения качества передачи микроволновых систем.
- Потребность в скоростной передаче информации является превалирующей, поэтому все современные системы беспроводной передачи информации работают в высокочастотных диапазонах, начиная с 800 МГц, несмотря на преимущества, которые сулят низкочастотные диапазоны благодаря распространению сигнала вдоль поверхности земли или отражения от ионосферы.
- Для успешного использования микроволнового диапазона необходимо также учитывать дополнительные проблемы, связанные с поведением сигналов, распространяющихся в режиме прямой видимости и встречающих на своем пути препятствия.

На рис. 10.4 показано, что сигнал, встретившись с препятствием, может распространяться в соответствии с тремя механизмами: отражением, дифракцией и рассеиванием.

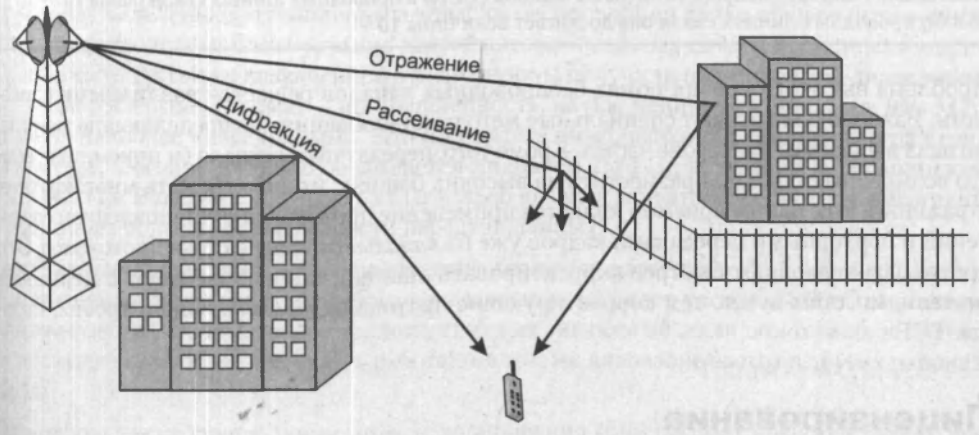


Рис. 10.4. Распространение электромагнитной волны

Когда сигнал встречается с препятствием, которое частично прозрачно для данной длины волны и в то же время размеры которого намного превышают длину волны, то часть

энергии сигнала *отражается* от такого препятствия. Волны микроволнового диапазона имеют длину несколько сантиметров, поэтому они частично отражаются от стен домов при передаче сигналов в городе. Если сигнал встречает непроницаемое для него препятствие (например, металлическую пластину) также намного большего размера, чем длина волны, то происходит **дифракция** — сигнал как бы огибает препятствие, так что такой сигнал можно получить, даже не находясь в зоне прямой видимости. И наконец, при встрече с препятствием, размеры которого соизмеримы с длиной волны, сигнал *рассеивается*, распространяясь под различными углами.

В результате подобных явлений, которые повсеместно встречаются при беспроводной связи в городе, приемник может получить несколько копий одного и того же сигнала. Такой эффект называется **многолучевым распространением сигнала**. Результат многолучевого распространения сигнала часто оказывается отрицательным, поскольку один из сигналов может прийти с обратной фазой и подавить основной сигнал.

Так как время распространения сигнала вдоль различных путей будет в общем случае различным, то может также наблюдаться **межсимвольная интерференция** — ситуация, когда в результате задержки сигналы, кодирующие соседние биты данных, доходят до приемника одновременно.

Искажения из-за многолучевого распространения приводят к ослаблению сигнала, этот эффект называется **многолучевым замиранием**. В городах многолучевое замирание приводит к тому, что ослабление сигнала становится пропорциональным не квадрату расстояния, а его кубу или даже четвертой степени!

Все эти искажения сигнала складываются с внешними электромагнитными помехами, которых в городе много. Достаточно сказать, что в диапазоне 2,4 ГГц работают микроволновые печи.

ВНИМАНИЕ

Отказ от проводов и обретение мобильности приводит к высокому уровню помех в беспроводных линиях связи. Если интенсивность битовых ошибок (BER) в проводных линиях связи равна 10^{-9}–10^{-10}, то в беспроводных линиях связи она достигает величины 10^{-3}!

Проблема высокого уровня помех беспроводных каналов решается различными способами. Важную роль играют специальные методы кодирования, распределяющие энергию сигнала в широком диапазоне частот. Кроме того, передатчики сигнала (и приемники, если это возможно) стараются разместить на высоких башнях, чтобы избежать многократных отражений. Еще одним приемом является применение протоколов с установлением соединений и повторными передачами кадров уже на *канальном* уровне стека протоколов. Эти протоколы позволяют быстрее корректировать ошибки, так как работают с меньшими значениями тайм-аутов, чем корректирующие протоколы *транспортного* уровня, такие как TCP.

Лицензирование

Итак, электромагнитные волны могут распространяться во всех направлениях на значительные расстояния и проходить через препятствия, такие как стены домов. Поэтому проблема разделения электромагнитного спектра является весьма острой и требует *централизованного* регулирования. В каждой стране есть специальный государственный орган,

который (в соответствии с рекомендациями ITU) выдает **лицензии** операторам связи на использование определенной части спектра, достаточной для передачи информации по определенной технологии. Лицензия выдается на определенную территорию, в пределах которой оператор использует закрепленный за ним диапазон частот монопольно.

При выдаче лицензий правительственные органы руководствуются различными стратегиями. Наиболее популярными являются три: конкурс, лотерея, аукцион.

- Участники **конкурса** — операторы связи — разрабатывают детальные предложения. В них они описывают свои будущие услуги, технологии, которые будут использоваться для реализации этих услуг, уровень цен для потенциальных клиентов и т. п. Затем комиссия рассматривает все предложения и выбирает оператора, который в наилучшей степени будет соответствовать общественным интересам. Сложность и неоднозначность критериев выбора победителя в прошлом часто приводили к значительным задержкам в принятии решений и коррупции среди государственных чиновников, поэтому некоторые страны, например США, отказались от такого метода. В то же время в других странах он все еще используется, чаще всего для наиболее значимых для страны услуг, например развертывания современных систем мобильной связи 3G.
- **Лотерея** — это наиболее простой способ, но он также не всегда приводит к справедливым результатам, поскольку в лотерее могут принимать участие и «подставные» операторы, которые собираются не вести операторскую деятельность, а просто перепродать лицензию.
- **Аукционы** сегодня являются достаточно популярным способом выявления обладателя лицензии. Они отсекают недобросовестные компании и приносят немалые доходы государствам. Впервые аукцион был проведен в Новой Зеландии в 1989 году. В связи с бумом вокруг мобильных систем 3G многие государства за счет подобных аукционов в значительной степени пополнили свои бюджеты.

Существуют также три частотных диапазона, 900 МГц, 2,4 ГГц и 5 ГГц, которые рекомендованы ITU как диапазоны для международного использования *без лицензирования*[1]. Эти диапазоны выделены промышленным товарам беспроводной связи общего назначения, например устройствам блокирования дверей автомобилей, научным и медицинским приборам. В соответствии с назначением эти диапазоны получили название **ISM-диапазонов** (Industrial, Scientific, Medical — промышленность, наука, медицина). Диапазон 900 МГц является наиболее «населенным». Это и понятно, низкочастотная техника всегда стоила дешевле. Сегодня активно осваивается диапазон 2,4 ГГц, например, в технологиях IEEE 802.11 и Bluetooth. Диапазон 5 ГГц только начал осваиваться, несмотря на то что он обеспечивает более высокие скорости передачи данных.

Обязательным условием использования этих диапазонов на совместной основе является ограничение максимальной мощности передаваемых сигналов уровнем 1 Вт. Это условие ограничивает радиус действия устройств, чтобы их сигналы не стали помехами для других пользователей, которые, возможно, работают в том же диапазоне частот в других районах города.

Существуют также специальные методы кодирования (они рассматриваются далее), позволяющие уменьшить взаимное влияние устройств, работающих в ISM-диапазонах.

---

[1] Диапазоны 900 МГц и 5 ГГц свободны от лицензирования не во всех странах.

Беспроводные системы

Двухточечная связь

Типичная схема проводного двухточечного канала является популярной и для беспроводной связи. По двухточечной схеме могут работать беспроводные каналы различного назначения, использующие различные диапазоны частот.

В телекоммуникационных первичных сетях такая схема уже долгое время применяется для создания так называемых **радиорелейных линий связи**. Такую линию образуют несколько башен, на которых установлены параболические направленные антенны (рис. 10.5). Каждая такая линия работает в микроволновом диапазоне на частотах в несколько гигагерц. Направленная антенна концентрирует энергию в узком пучке, что позволяет передавать информацию на значительные расстояние, обычно до 50 км. Высокие башни обеспечивают прямую видимость антенн.

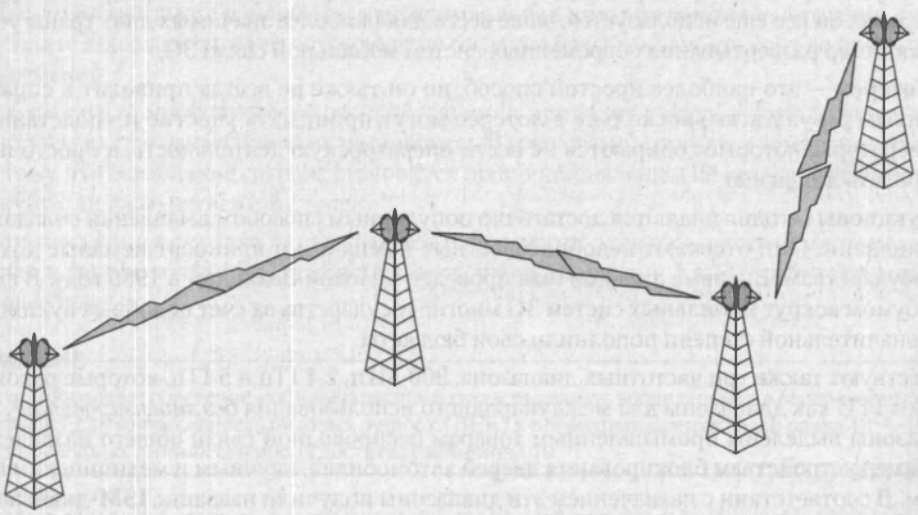


Рис. 10.5. Радиорелейная линия связи

Пропускная способность линии может быть достаточно высокой, обычно она находится в пределах от нескольких до сотен мегабитт в секунду. Это могут быть как магистральные линии, так и линии доступа (в последнем случае они имеют чаще всего один канал). Операторы связи часто используют подобные линии, когда прокладка оптического волокна либо невозможна (из-за природных условий), либо экономически невыгодна.

Радиорелейная линия связи может использоваться в городе для соединения двух зданий. Так как высокая скорость в таком случае не всегда нужна (например, нужно соединить небольшой сегмент локальной сети с основной локальной сетью предприятия), то здесь могут применяться радиомодемы, работающие в АМ-диапазоне. Для связи двух зданий может также использоваться лазер, обеспечивая высокую информационную скорость (до 155 Мбит/с), но только при соответствующем состоянии атмосферы.

Другой пример беспроводной двухточечной линии связи показан на рис. 10.6. Здесь она служит для соединения двух компьютеров. Такая линия образует простейший сегмент локальной сети, поэтому расстояния и мощности сигнала здесь принципиально иные.

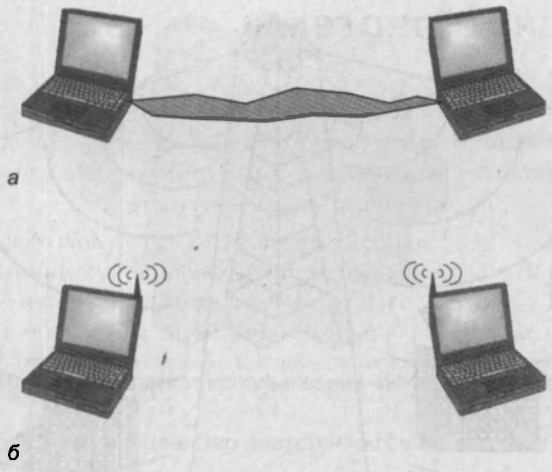


Рис. 10.6. Беспроводная связь двух компьютеров

Для расстояний в пределах одного помещения может использоваться диапазон инфракрасных волн (рис. 10.6, *а*) или микроволновый диапазон (рис. 10.6, *б*). Большинство современных ноутбуков оснащено встроенным инфракрасным портом, поэтому такое соединение может быть образовано автоматически, как только порты двух компьютеров окажутся в пределах прямой видимости (или видимости отраженного луча).

Микроволновый вариант работает в пределах нескольких десятков или сотен метров — предельное расстояние предсказать невозможно, так как при распространении микроволнового сигнала в помещении происходят многочисленные отражения, дифракции и рассеивания, к которым добавляются эффекты проникновения волн через стены и межэтажные перекрытия.

Связь одного источника и нескольких приемников

Схема беспроводного канала с одним источником и несколькими приемниками характерна для такой организации доступа, при которой многочисленные пользовательские терминалы соединяются с **базовой станцией** (Base Station, BS).

Беспроводные линии связи в схеме с одним источником и несколькими приемниками служат как для фиксированного доступа, так и для мобильного.

На рис. 10.7 показан вариант фиксированного доступа с помощью микроволновых линий связи. Оператор связи использует высокую башню (возможно, телевизионную), чтобы обеспечить прямую видимость с антеннами, установленными на крышах зданий своих клиентов. Фактически такой вариант может представлять собой набор двухточечных линий связи — по количеству зданий, которые необходимо соединить с базовой станцией. Однако это достаточно расточительный вариант, так как для каждого нового клиента нужно устанавливать новую антенну на башне. Поэтому для экономии обычно применяют антенны, захватывающие определенный сектор, например, в 45°. Тогда за счет нескольких антенн оператор может обеспечить связь в пределах полного сектора в 360°, конечно, на ограниченном расстоянии (обычно несколько километров).

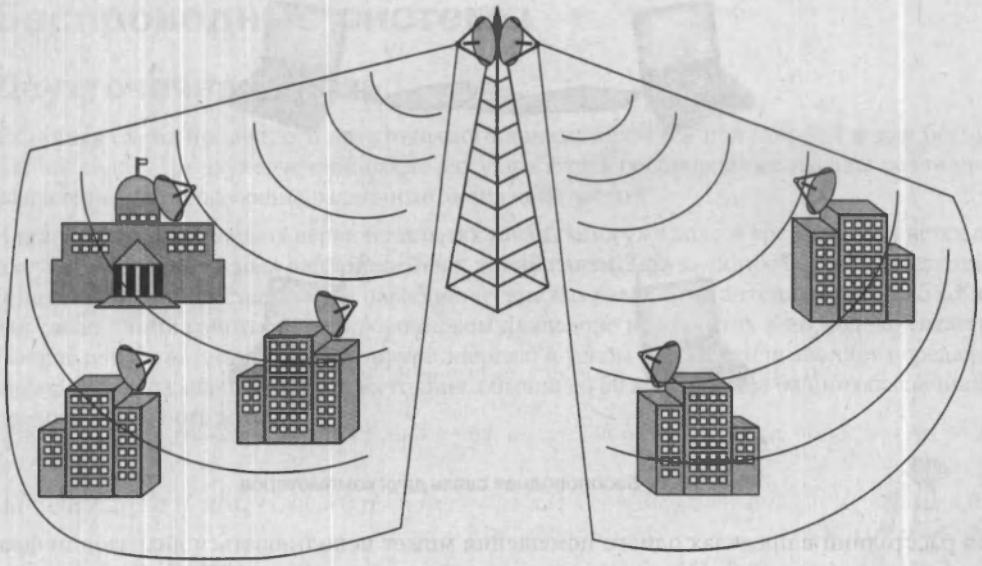


Рис. 10.7. Фиксированный беспроводный доступ

Пользователи линий доступа могут обмениваться информацией только с базовой станцией, а она, в свою очередь, транзитом обеспечивает взаимодействие между отдельными пользователями.

Базовая станция обычно соединяется проводной связью с проводной частью сети, обеспечивая взаимодействие с пользователями других базовых станций или пользователями проводных сетей. Поэтому базовая станция также называется **точкой доступа** (Access Point, AP). Точка доступа включает не только оборудование DCE, необходимое для образования линии связи, но и чаще всего является коммутатором сети, доступ к которой она обеспечивает — телефонным коммутатором или коммутатором пакетов.

В большинстве схем мобильного доступа используется сегодня принцип **сот**, которые представляют собой небольшие по площади территории, обслуживаемые одной базовой станцией. Идея сот родилась не сразу, первые мобильные телефоны работали по другому принципу, обращаясь к одной базовой станции, покрывающей большую территорию. Идея небольших сот была впервые сформулирована еще в 1945 году, с тех пор прошло довольно много времени, пока заработали первые коммерческие сотовые телефонные сети — пробные участки появились в конце 60-х, а широкое коммерческое применение началось в начале 80-х.

Принцип разбиения всей области охвата сети на небольшие соты дополняется идеей многократного использования частоты. На рис. 10.8 показан вариант организации сот при наличии всего трех частот, при этом ни одна из соседних пар сот не задействует одну и ту же частоту. Многократное использование частот позволяет оператору экономно расходовать выделенный ему частотный диапазон, при этом абоненты и базовые станции соседних сот не испытывают проблем из-за интерференции сигналов. Конечно, базовая станция должна контролировать мощность излучаемого сигнала, чтобы две соты (несмежные), работающие на одной и той же частоте, не создавали друг другу помех.

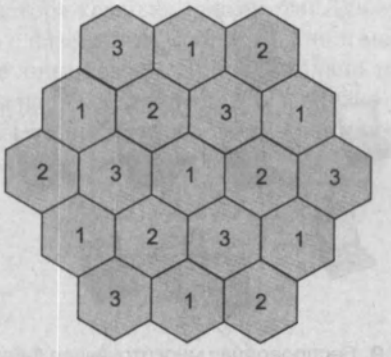


Рис. 10.8. Многократное использование частот в сотовой сети

При гексагональной форме сот количество повторяемых частот может быть больше, чем 3, например 4, 7, 9, 12, 13 и т. д.

Если известно минимальное расстояние D между центрами сот, работающих на одной и той же частоте, то число сот (N) можно выбрать по формуле:

$$N = D^2/3R^2,$$

где R – радиус соты.

Небольшие по величине соты обеспечивают небольшие габариты и мощность терминального устройства пользователя. Именно это обстоятельство (а также общий технологический прогресс) позволяет современным мобильным телефонам быть такими компактными.

Мобильные компьютерные сети пока не получили такого распространения, как телефонные, но принципы организации беспроводных линий связи в них остаются теми же.

Важной проблемой мобильной линии связи является переход терминального устройства из одной соты в другую. Эта процедура, которая называется **эстафетной передачей**, отсутствует при фиксированном доступе и относится к протоколам более высоких уровней, нежели физический.

Связь нескольких источников и нескольких приемников

В случае схемы с несколькими источниками и несколькими приемниками беспроводная линия связи представляет собой общую электромагнитную среду, разделяемую несколькими узлами. Каждый узел может использовать эту среду для взаимодействия с любым другим узлом без обращения к базовой станции. Так как базовая станция отсутствует, то необходим децентрализованный алгоритм доступа к среде.

Чаще всего такой вариант беспроводного канала применяется для соединения компьютеров (рис. 10.9). Для телефонного трафика неопределенность в доле пропускной способности, получаемой при разделении среды, может резко ухудшить качество передачи голоса. Поэтому они строятся по ранее рассмотренной схеме с одним источником (базовой станцией) для распределения полосы пропускания и несколькими приемниками.

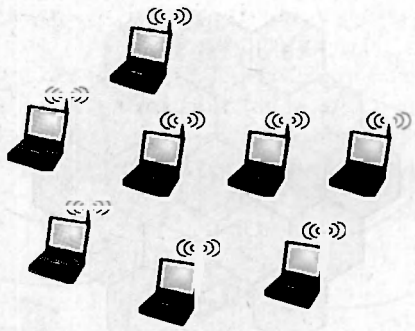


Рис. 10.9. Беспроводная многоточечная линия связи

Собственно, первая локальная сеть, созданная в 70-е годы на Гавайях, в точности соответствовала приведенной на рисунке схеме. Ее отличие от современных беспроводных локальных сетей состоит в низкой скорости передачи данных (9600 бит/с), а также в весьма неэффективном способе доступа, позволяющем использовать только 18 % полосы пропускания.

Сегодня подобные сети передают данные со скоростью до 52 Мбит/с[1] в микроволновом или инфракрасном диапазоне. Для связи каждого с каждым служат ненаправленные антенны. Для того чтобы инфракрасный свет распространялся в разных направлениях, применяются **диффузные передатчики**, которые рассеивают лучи с помощью системы линз.

Типы спутниковых систем

Спутниковая связь используется для организации высокоскоростных микроволновых протяженных линий. Так как для таких линий связи нужна прямая видимость, которую из-за кривизны Земли невозможно обеспечить на больших расстояниях, то спутник как отражатель сигнала является естественным решением этой проблемы (рис. 10.10).

Идея задействовать искусственный спутник Земли для создания линий связи родилась задолго до запуска в 1957 году первого такого спутника Советским Союзом. Писатель-фантаст Артур Кларк продолжил дело Жюля Верна и Герберта Уэллса, которым удалось описать множество технических изобретений еще до их появления. Кларк в 1945 году описал геостационарный спутник, который висит над одной точкой экватора и обеспечивает связью большую территорию Земли.

Первый спутник, запущенный Советским Союзом в годы холодной войны, обладал очень ограниченными телекоммуникационными возможностями — он только передавал радиосигнал «бин-бип», извещая мир о своем присутствии в космосе. Однако успех России в космосе подхлестнул усилия Америки, и в 1962 году она запустила первый телекоммуникационный спутник Telstar-1, который поддерживал 600 голосовых каналов.

Со времени запуска первого телекоммуникационного спутника прошло уже более 40 лет, и функции спутника как телекоммуникационного узла, естественно, усложнились. Сегодня спутник может играть роль узла первичной сети, а также телефонного коммутатора

[1] Новая версия стандарта беспроводных локальных сетей предусматривает повышение скорости передачи данных до 300 Мбит/с.

и коммутатора/маршрутизатора компьютерной сети. Для этого аппаратура спутников взаимодействует не только с наземными станциями, но и между собой, образуя прямые космические беспроводные линии связи. Принципиально техника передачи микроволновых сигналов в космосе и на Земле не отличается, однако у спутниковых линий связи есть и очевидная специфика — один из узлов такой линии постоянно находится в полете, причем на большом расстоянии от других узлов.

Рис. 10.10. Спутник как отражатель сигнала

Для спутниковой связи союз ITU выделил несколько частотных диапазонов (табл. 10.1).

Таблица 10.1. Частотные диапазоны спутниковой связи

Диапазон	Нисходящая частота, ГГц	Восходящая частота, ГГц
L	1,5	1,6
S	1,9	2,2
C	3,7–4,2	5,925–6,425
Ku	11,7–12,2	14,0–14,5
Ka	17,7–21,7	27,5–30,5

Исторически первым использовался диапазон **C**, в котором для каждого из дуплексных потоков Земля-спутник (восходящая частота) и спутник-Земля (нисходящая частота) выделяется по 500 МГц — этого достаточно для большого числа каналов. Диапазоны **L** и **S** предназначаются для организации мобильных услуг с помощью спутников. Они также часто используются наземными системами. Диапазоны **Ku** и **Ka** пока мало «населены» на Земле, их применению препятствует высокая стоимость оборудования, особенно для диапазона **Ka**.

Искусственные спутники Земли вращаются вокруг нее в соответствии с законами, открытыми Йоханесом Кеплером (Johannes Kepler). Орбита вращения спутника в общем случае является эллиптической, но для сохранения постоянной высоты над Землей спутники могут переходить на почти круговую орбиту.

Сегодня используют три группы круговых орбит, отличающихся высотой над Землей (рис. 10.11):

- геостационарная орбита (Geostationary Orbit, GEO) — 35 863 км;
- средневысотная орбита (Medium Earth Orbit, MEO) — 5000–15 000 км;
- маловысотная орбита (Low Earth Orbit, LEO) — 100–1000 км.

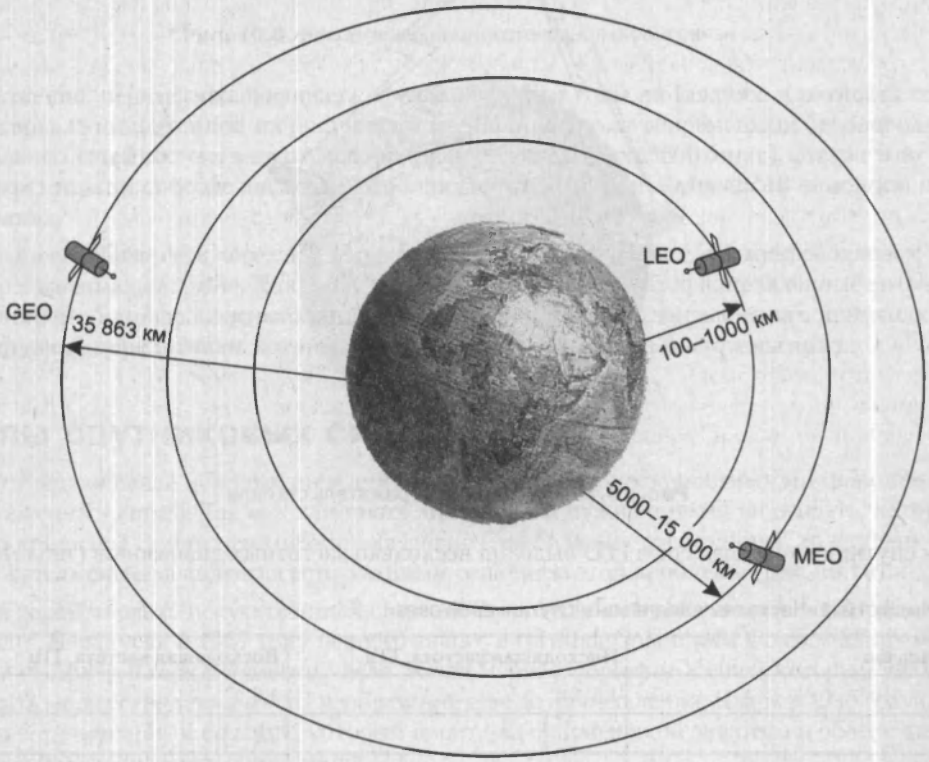


Рис. 10.11. Типы орбит спутников

Геостационарный спутник

Геостационарный спутник «висит» над определенной точкой экватора, в точности следуя скорости вращения Земли. Такое положение выгодно по следующим обстоятельствам.

Во-первых, четверть поверхности Земли оказывается с такой высоты в зоне прямой видимости, поэтому с помощью геостационарных спутников *просто организовать широковещание в пределах страны или даже континента.*

Во-вторых, сам *спутник неподвижен для наземных антенн*, что значительно облегчает организацию связи, так как не нужно автоматически корректировать направление наземной антенны, как это приходится делать для низкоорбитальных и средневысотных спутников. Правда, с появлением в 1990 небольших всенаправленных антенн ситуация изменилась — теперь уже не нужно следить за положением низкоорбитального спутника, достаточно, чтобы он находился в зоне прямой видимости.

В-третьих, геостационарный спутник находится за пределами земной атмосферы и *меньше «изнашивается»*, чем низкоорбитальные и средневысотные спутники. Низкоорбитальные спутники из-за трения о воздух постоянно теряют высоту и им приходится восстанавливать ее с помощью двигателей.

Путем применения нескольких антенн геостационарные спутники обычно *поддерживают большое количество каналов*. Раньше для работы с геостационарными спутниками в качестве антенн требовались очень большие тарелки (диаметром до 10 м). Это затрудняло использование геостационарных спутников для небольших организаций и личных целей. Однако ситуация изменилась с появлением направленных антенн, устанавливаемых на спутниках. Такие антенны создают сигнал, который можно принимать с помощью сравнительно небольших наземных антенн, так называемых миниатюрных апертурных терминалов (Very Small Aperture Terminals, VSAT). Диаметр антенны VSAT составляет около 1 м. Наземные станции, оснащенные VSAT, предоставляют сегодня широкий набор услуг, к которым относятся телефония, передача данных, конференции.

Наряду с достоинствами у геостационарных спутников есть и недостатки. Наиболее очевидные связаны с *большим удалением спутника от поверхности Земли*. Это приводит к большим задержкам распространения сигнала — от 230 до 280 мс. При использовании спутника для передачи разговора или телевизионного диалога возникают неудобные паузы, мешающие нормальному общению.

Кроме того, на таких расстояниях *потери сигнала высоки*, что означает необходимость применения мощных передатчиков и тарелок больших размеров (это не относится к антеннам VSAT, но при их использовании уменьшается область охвата).

Принципиальным недостатком геостационарного спутника с его круговой орбитой является также *плохая связь для районов, близких к Северному и Южному полюсам*. Сигналы в таких районах проходят большие расстояния, чем в районах, расположенных в экваториальных и умеренных широтах, и, естественно, больше ослабляются. Решением является спутник с ярко выраженной эллиптической орбитой, который приближается к Земле как раз в районе Северного и Южного полюсов. Примером такого спутника являются спутники серии «Молния», которые запускаются Россией, имеющей большие территории на Крайнем Севере.

Место на орбите геостационарного спутника также регулируется союзом ITU. Сегодня наблюдается определенный дефицит таких мест, так как геостационарные спутники не могут располагаться на орбите ближе, чем 2° друг к другу. Из этого следует, что на орбите может находиться не более 180 геостационарных спутников. Так как не все страны в состоянии (пока) запустить геостационарный спутник, то здесь наблюдается та же ситуация, что и в конкурсе на получение определенного диапазона частот, только еще усиленная политическими амбициями стран.

Средне- и низкоорбитальные спутники

Класс среднеорбитальных спутников пока не так популярен, как геостационарных и низкоорбитальных. **Среднеорбитальные спутники** обеспечивают диаметр покрытия от 10 000 до 15 000 км и задержку распространения сигнала 50 мс. Наиболее известной услугой, предоставляемой спутниками этого класса, является *глобальная система навигации* (Global Positioning System, GPS), известная также под названием NAVigation Satellites providing Time And Range (NAVSTAR). GPS — это всеобщая система определения текущих координат пользователя на поверхности Земли или в околоземном пространстве. GPS состоит из 24 спутников — это то минимальное число спутников, которое необходимо для 100-процентного покрытия территории Земли. Первый тестовый спутник GPS был запущен в 1974 году, первый промышленный спутник — в 1978 году, а 24-й промышленный — в 1993 году. Спутники GPS летают на орбите высотой около 20 000 км. Помимо спутников в систему GPS входит сеть наземных станций слежения за ними и неограниченное количество пользовательских приемников-вычислителей, среди которых и ставшие очень популярными в последние годы приемники автомобильных систем навигации.

По радиосигналам спутников GPS-приемники пользователей устойчиво и точно определяют координаты; для этого на поверхности Земли приемнику необходимо принять сигналы как минимум от трех спутников. Погрешности не превышают десятков метров. Этого вполне достаточно для решения задач навигации подвижных объектов (самолеты, корабли, космические аппараты, автомобили и т. д.).

В СССР была разработана и реализована система аналогичного назначения под названием ГЛОНАСС (*ГЛО*бальная *НА*вигационная *С*путниковая *С*истема). Первый спутник ГЛОНАСС был запущен в октябре 1982 года, а в сентябре 1993 года система была официально введена в эксплуатацию. В 1995 году количество спутников достигло плановой цифры 24, но затем из-за проблем с финансированием не все выходившие из строя спутники заменялись новыми, поэтому было время, когда их число уменьшилось до 14, хотя в декабре 2008 количество спутников удалось увеличить до 18. Система ГЛОНАСС совместима с GPS, существует навигационное оборудование, которое может принимать сигналы от спутников обеих систем.

Достоинства и недостатки **низкоорбитальных спутников** противоположны соответствующим качествам геостационарных спутников. Главное их достоинство — близость к Земле, а значит, пониженная мощность передатчиков, малые размеры антенн и небольшое время распространения сигнала (около 20–25 мс). Кроме того, их легче запускать. Основной недостаток — малая площадь покрытия, диаметр которой составляет всего около 8000 км. Период оборота такого спутника вокруг Земли составляет 1,5–2 часа, а время видимости спутника наземной станцией — всего 20 минут. Это значит, что постоянная связь с помощью низкоорбитальных спутников может быть обеспечена, только когда на орбите находится достаточно большое их количество. Кроме того, атмосферное трение снижает срок службы таких спутников до 8–10 лет.

Если основным назначением геостационарных спутников является широковещание и дальняя связь, то низкоорбитальные спутники рассматриваются как важное средство поддержания мобильной связи.

В начале 90-х годов достоинства компактных терминальных устройств для низкоорбитальных спутников показались руководителям компании Motorola более важными, чем их недостатки. Вместе с несколькими крупными партнерами эта компания начала про-

ект *Indium*, который имел весьма амбициозную цель — создать всемирную спутниковую сеть, обеспечивающую мобильную связь в любой точке земного шара. В конце 80-х еще не существовало такой плотной системы сот мобильной телефонии, как сегодня, так что коммерческий успех казался обеспеченным.

В 1997 группа из 66 спутников была запущена, а в 1998 году началась коммерческая эксплуатация системы Iridium. Спутники Iridium действительно покрывают всю поверхность земного шара, вращаясь по 6 орбитам, проходящим через полюсы Земли. На каждой орбите находится по 11 спутников, передатчики которых работают на частоте 1,6 ГГц с полосой пропускания 10 МГц. Эта полоса расходуется 240 каналами по 41 кГц каждый. За счет многократного использования частот система Iridium поддерживает 253 440 каналов, организуя системы скользящих по поверхности Земли сот. Для пользователей системы Iridium основным видом услуги является телефонная связь и передача данных со скоростью 2,4 Кбит/с.

Спутники Iridium обладают значительным интеллектом, они могут, пользуясь специальными межспутниковыми каналами, передавать друг другу информацию со скоростью 25 Мбит/с. Поэтому телефонный вызов идет от спутникового телефона Iridium прямо на спутник, находящийся в зоне видимости. Затем этот спутник маршрутизирует вызов через систему промежуточных спутников тому спутнику, который в данный момент ближе к вызываемому абоненту. Система Iridium представляет собой сеть с полным собственным стеком протоколов, поддерживающим всемирный роуминг.

К сожалению, коммерческие успехи Iridium оказались очень скромными, и через два года своего существования компания обанкротилась. Расчет на мобильных телефонных абонентов оказался неверным — к моменту начала работы наземная сеть сотовой связи уже покрывала большую часть территории развитых стран. А услуги по передаче данных со скоростью 2,4 Кбит/с не соответствовали потребностям пользователей конца XX века.

Сегодня система Iridium снова работает, теперь уже с новым владельцем и новым именем — *Iridium Satellite*. У нее теперь более скромные планы, связанные с созданием местных систем связи в тех частях земного шара, где другая связь практически отсутствует. Программное обеспечение спутников модернизируется «на лету», что позволило повысить скорость передачи данных до 10 Кбит/с. В феврале 2008 года компания Iridium Satellite объявила о новой программе под названием Iridium NEXT. В соответствии с этой программой к 2014 году будут запущены новые 66 спутников; все коммуникации со спутниками и между спутниками будут происходить на основе стека протоколов TCP/IP.

Другой известной системой низкоорбитальных спутников является *Globalstar*. В отличие от Iridium 48 низкоорбитальных спутников Globalstar выполняют традиционные для геостационарных спутников функции — принимают телефонные вызовы от мобильных абонентов и передают их ближайшей наземной базовой станции. Маршрутизацию вызовов выполняет базовая станция, перенаправляющая вызов базовой станции, ближайшей к спутнику, в зоне видимости которого находится вызываемый абонент. Межспутниковые каналы не используются. Помимо телефонных разговоров Globalstar передает данные со скоростью 4,8 Кбит/с.

Еще одна сеть LEO — *Orbcomm* предоставляет сервис, ориентированный на передачу коротких сообщений в режиме «машина-машина», например, между промышленными установками или датчиками, расположенными в труднодоступных районах. Доставка сообщений не всегда осуществляется в режиме реального времени. Если спутник невидим, терминал Orbcomm просто хранит пакеты, пока космический аппарат не войдет в зону видимости.

Это приводит к чрезвычайно значительной неравномерности в передаче данных. Вместо привычных для пользователей Интернета задержек в доли секунды, в этой сети паузы иногда измеряются минутами.

Технология широкополосного сигнала

Техника расширенного спектра разработана специально для беспроводной передачи. Она позволяет повысить помехоустойчивость кода для сигналов малой мощности, что очень важно в мобильных приложениях. Однако нужно подчеркнуть, что техника расширенного спектра — не единственная техника кодирования, которая применяется для беспроводных линий связи микроволнового диапазона. Здесь также применяются частотная (FSK) и фазовая (PSK) манипуляции, описанные в предыдущей главе. Амплитудная манипуляция (ASK) не используется по той причине, что каналы микроволнового диапазона имеют широкую полосу пропускания, а усилители, которые обеспечивают одинаковый коэффициент усиления для широкого диапазона частот, очень дороги.

Широкая полоса пропускания позволяет также применять модуляцию с несколькими несущими, когда полоса делится на несколько подканалов, каждый из которых имеет собственную несущую частоту. Соответственно, битовый поток делится на несколько подпотоков, текущих с более низкой скоростью. Затем каждый подпоток модулируется с помощью определенной несущей частоты, которая обычно кратна основной несущей частоте, то есть f_0, $2f_0$, $3f_0$ и т. д. Модуляция выполняется с помощью обычных методов FSK или PSK. Такая техника называется **ортогональным частотным мультиплексированием** (Orthogonal Frequency Division Multiplexing, OFDM).

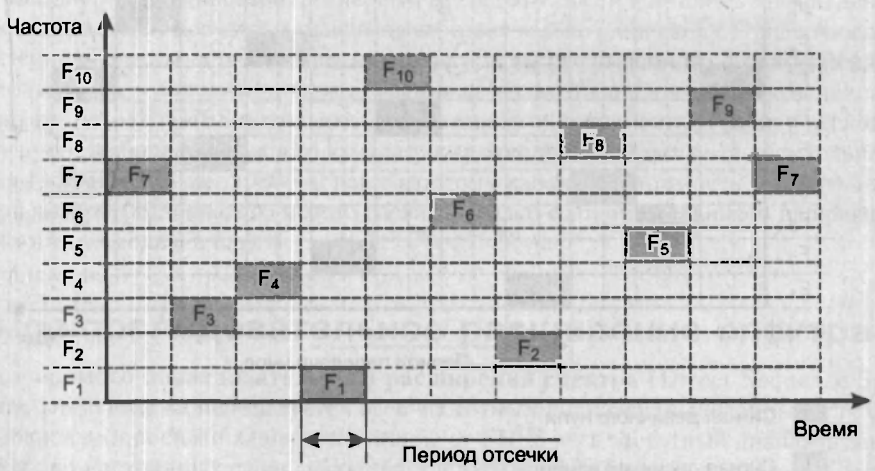
Перед передачей все несущие сворачиваются в общий сигнал путем быстрого преобразования Фурье. Спектр такого сигнала примерно равен спектру сигнала, кодируемого одной несущей. После передачи из общего сигнала путем обратного преобразования Фурье выделяются несущие подканалы, а затем из каждого канала выделяется битовый поток. Выигрыш в разделении исходного высокоскоростного битового потока на несколько низкоскоростных подпотоков проявляется в том, что увеличивается интервал между отдельными символами кода. Это означает, что снижается эффект межсимвольной интерференции, появляющийся из-за многолучевого распространения электромагнитных волн.

Расширение спектра скачкообразной перестройкой частоты

Идея метода **расширения спектра скачкообразной перестройкой частоты** (Frequency Hopping Spread Spectrum, FHSS) возникла во время Второй мировой войны, когда радио широко использовалось для секретных переговоров и управления военными объектами, например торпедами. Для того чтобы радиообмен нельзя было перехватить или подавить узкополосным шумом, было предложено вести передачу с постоянной сменой несущей в пределах широкого диапазона частот. В результате мощность сигнала распределялась по всему диапазону, и прослушивание какой-то определенной частоты давало только небольшой шум. Последовательность несущих частот выбиралась псевдослучайной, известной только передатчику и приемнику. Попытка подавления сигнала в каком-то узком

диапазоне также не слишком ухудшала сигнал, так как подавлялась только небольшая часть информации.

Идею этого метода иллюстрирует рис. 10.12.



Последовательность перестройки частот: F_7-F_3-F_4-F_1-F_{10}-F_6-F_2-F_8-F_5-F_9

Рис. 10.12. Расширение спектра скачкообразной перестройкой частоты

В течение определенного фиксированного интервала времени передача ведется на неизменной несущей частоте. На каждой несущей частоте для передачи дискретной информации применяются стандартные методы модуляции, такие как FSK или PSK. Чтобы приемник синхронизировался с передатчиком, для обозначения начала каждого периода передачи в течение некоторого времени передаются синхробиты. Так что полезная скорость этого метода кодирования оказывается меньше из-за постоянных накладных расходов на синхронизацию.

Несущая частота меняется в соответствии с номерами частотных подканалов, вырабатываемых алгоритмом псевдослучайных чисел. Псевдослучайная последовательность зависит от некоторого параметра, который называют **начальным числом**. Если приемнику и передатчику известны алгоритм и значение начального числа, то они меняют частоты в одинаковой последовательности, называемой **последовательностью псевдослучайной перестройки частоты**.

Если частота смены подканалов ниже, чем скорость передачи данных в канале, то такой режим называют **медленным расширением спектра** (рис. 10.13, *а*); в противном случае мы имеем дело с **быстрым расширением спектра** (рис. 10.13, *б*).

Метод быстрого расширения спектра более устойчив к помехам, поскольку узкополосная помеха, которая подавляет сигнал в определенном подканале, не приводит к потере бита, так как его значение повторяется несколько раз в различных частотных подканалах. В этом режиме не проявляется эффект межсимвольной интерференции, потому что ко времени прихода задержанного вдоль одного из путей сигнала система успевает перейти на другую частоту.

а

б

Рис. 10.13. Соотношение между скоростью передачи данных и частотой смены подканалов

Метод медленного расширения спектра таким свойством не обладает, но зато он проще в реализации и имеет меньшие накладные расходы.

Методы FHSS применяют в беспроводных технологиях IEEE 802.11 и Bluetooth.

В методах FHSS подход к использованию частотного диапазона не такой, как в других методах кодирования — вместо экономного расходования узкой полосы делается попытка занять весь доступный диапазон. На первый взгляд это кажется не очень эффективным — ведь в каждый момент времени в диапазоне работает только один канал. Однако последнее утверждение не всегда справедливо, поскольку коды расширенного спектра можно задействовать также и для мультиплексирования *нескольких* каналов в широком диапазоне. В частности, методы FHSS позволяют организовать одновременную работу нескольких каналов путем выбора для каждого канала таких псевдослучайных последовательностей, которые в каждый момент времени дают каждому каналу возможность работать на собственной частоте (конечно, это можно сделать, только если число каналов не превышает числа частотных подканалов).

Прямое последовательное расширение спектра

В методе **прямого последовательного расширения спектра** (Direct Sequence Spread Spectrum, DSSS) также используется весь частотный диапазон, выделенный для одной беспроводной линии связи. Однако в отличие от FHSS весь частотный диапазон занимается не за счет постоянных переключений с частоты на частоту, а за счет того, что каждый бит информации заменяется N битами, поэтому тактовая скорость передачи сигналов увеличивается в N раз. А это, в свою очередь, означает, что спектр сигнала также расширяется в N раз. Достаточно соответствующим образом выбрать скорость передачи данных и значение N, чтобы спектр сигнала заполнил весь диапазон.

Цель кодирования методом DSSS та же, что методом FHSS — повышение помехоустойчивости. Узкополосная помеха будет искажать только определенные частоты спектра сигнала, так что приемник с большой степенью вероятности сможет правильно распознать передаваемую информацию.

Код, которым заменяется двоичная единица исходной информации, называется **расширяющей последовательностью**, а каждый бит такой последовательности — **чипом**. Соответственно, скорость передачи результирующего кода называют **чиповой скоростью**. Двоичный нуль кодируется инверсным значением расширяющей последовательности. Приемники должны знать расширяющую последовательность, которую использует передатчик, чтобы понять передаваемую информацию.

Количество битов в расширяющей последовательности определяет **коэффициент расширения** исходного кода. Как и в случае FHSS, для кодирования битов результирующего кода может использоваться любой вид модуляции, например BFSK.

Чем больше коэффициент расширения, тем шире спектр результирующего сигнала и тем больше степень подавления помех. Но при этом растет занимаемый каналом диапазон спектра. Обычно коэффициент расширения имеет значения от 10 до 100.

Примером расширяющей последовательности является *последовательность Баркера* (Barker), которая состоит из 11 бит: 10110111000. Если передатчик использует эту последовательность, то передача трех битов 110 ведет к отправке следующих битов:

10110111000 10110111000 01001000111.

Последовательность Баркера позволяет приемнику быстро синхронизироваться с передатчиком, то есть надежно выявлять начало последовательности. Приемник определяет

такое событие, поочередно сравнивая получаемые биты с образцом последовательности. Действительно, если сравнить последовательность Баркера с такой же последовательностью, но сдвинутой на один бит влево или вправо, то мы получим меньше половины совпадений значений битов. Значит, даже при искажении нескольких битов с большой долей вероятности приемник правильно определит начало последовательности, а значит, сможет правильно интерпретировать получаемую информацию.

Метод DSSS в меньшей степени защищен от помех, чем метод быстрого расширения спектра, так как мощная узкополосная помеха влияет на часть спектра, а значит, и на результат распознавания единиц или нулей.

Множественный доступ с кодовым разделением

Как и в случае FHSS, кодирование методом DSSS позволяет мультиплексировать несколько каналов в одном диапазоне. Техника такого мультиплексирования называется **множественным доступом с кодовым разделением** (Code Division Multiplexing Access, CDMA). Она широко используется в сотовых сетях.

Хотя техника CDMA может применяться совместно с кодированием методом FHSS, на практике в беспроводной сети она чаще сочетается с методом DSSS.

Каждый узел сети, работающий по методу CDMA, посылает данные в разделяемую среду в те моменты времени, когда это ему нужно, то есть синхронизация между узлами отсутствует. Идея CDMA заключается в том, что каждый узел сети задействует собственное значение расширяющей последовательности. Эти значения выбираются так, чтобы принимающий узел, который знает значение расширяющей последовательности передающего узла, мог выделить данные передающего узла из суммарного сигнала, образующегося в результате одновременной передачи информации несколькими узлами.

Для того чтобы такую операцию демультиплексирования можно было выполнить, значения расширяющей последовательности выбираются определенным образом. Поясним идею CDMA на примере.

Пусть в сети работает четыре узла: *A*, *B*, *C* и *D*. Каждый узел использует следующие значения расширяющей последовательности:

A: 0 1 0 1 0 1 0 1
B: 1 0 1 0 0 1 0 1
C: 1 0 0 1 1 0 0 1
D: 1 1 1 1 1 1 1 1

Предположим также, что при передаче единиц и нулей расширяющей последовательности (то есть уже преобразованного исходного кода) используются сигналы, которые являются аддитивными и инверсными. Инверсность означает, что двоичная единица кодируется, например, синусоидой с амплитудой $+A$, а двоичный ноль — синусоидой с амплитудой $-A$. Из условия аддитивности следует, что если фазы этих амплитуд совпадут, то при одновременной передаче единицы и нуля мы получим нулевой уровень сигнала. Для упрощения записи расширяющей последовательности обозначим синусоиду с положительной амплитудой значением +1, а синусоиду с отрицательной амплитудой — значением –1. Для простоты допустим также, что все узлы сети CDMA синхронизированы.

Таким образом, при передаче единицы исходного кода 4 узла передают в среду такие последовательности:

A: –1 +1 –1 +1–1 +1 –1 +1
B: +1 –1 +1 –1 –1 +1 –1 +1
C: +1 –1 –1 +1 +1 –1 –1 +1
D: +1 +1 +1 +1 +1 +1 +1 +1

При передаче нуля исходного кода сигналы расширяющей последовательности инвертируются.

Пусть теперь каждый из 4-х узлов независимо от других передает в сеть один бит исходной информации: узел $A \rightarrow 1$, узел $B \rightarrow 0$, узел $C \rightarrow 0$, узел $D \rightarrow 1$.

В среде *S* сети наблюдается такая последовательность сигналов:

A: –1 +1 –1 +1–1 +1 –1 +1
B: –1 +1 –1 +1 +1 –1 +1 –1
C: –1 +1 +1 –1 –1 +1 +1 –1
D: +1 +1 +1 +1 +1 +1 +1 +1

В соответствии со свойством аддитивности получаем:

S: –2 +4 0 +2 0 +2 +2 0

Если, например, некоторый узел *E* хочет принимать информацию от узла *A*, то он должен использовать свой демодулятор CDMA, задав ему в качестве параметра значение расширяющей последовательности узла *A*.

Демодулятор CDMA последовательно складывает все четыре суммарных сигнала S_i, принятые в течение каждого такта работы. При этом сигнал S_i, принятый в такте, на котором код расширения станции *A* равен +1, учитывается в сумме со своим знаком, а сигнал, принятый в такте, на котором код расширения станции *A* равен –1, добавляется в сумму с противоположным знаком. Другими словами, демодулятор выполняет операцию скалярного умножения вектора принятых сигналов на вектор значения расширяющей последовательности нужной станции:

$$S \times A = (-2\ +4\ 0\ +2\ 0\ +2\ +2\ 0) \times (-1\ +1\ -1\ +1-1\ +1\ -1\ +1) = 8.$$

Для того чтобы узнать, какой бит послала станция *A*, осталось нормализовать результат, то есть разделить его на количество разрядов в расширяющей последовательности: 8/8 = 1. Если бы станция хотела принимать информацию от станции *B*, то ей нужно было бы при демодуляции использовать код расширения станции *B* (+1 –1 +1 –1 –1 +1 –1 +1):

$$S \times B = (-2\ +4\ 0\ +2\ 0\ +2\ +2\ 0) \times (+1\ -1\ +1\ -1\ -1\ +1\ -1\ +1) = -8.$$

После нормализации мы получаем сигнал –1, который соответствует двоичному нулю исходной информации станции *B*.

Мы объяснили только основную идею CDMA, предельно упростив ситуацию. На практике CDMA является весьма сложной технологией, которая оперирует не условными значениями +1 и –1, а модулированными сигналами, например сигналами BPSK. Кроме того, узлы сети не синхронизированы между собой, а сигналы, которые приходят от удаленных на различные расстояния от приемника узлов, имеют разную мощность. Проблема синхронизации приемника и передатчика решается за счет передачи длинной последовательности определенного кода, называемого **пилотным сигналом**. Для того же, чтобы мощности всех передатчиков были примерно равны для базовой станции, в CDMA применяются специальные процедуры управления мощностью.

Выводы

Беспроводная связь делится на мобильную и фиксированную. Для организации мобильной связи беспроводная среда является единственной альтернативой. Фиксированная беспроводная связь обеспечивает доступ к узлам сети, расположенным в пределах небольшой территории, например здания.

Каждый узел беспроводной линии связи оснащается антенной, которая одновременно является передатчиком и приемником электромагнитных волн.

Электромагнитные волны могут распространяться во всех направлениях или же в пределах определенного сектора. Тип распространения зависит от типа антенны.

Беспроводные системы передачи данных делятся на четыре группы в зависимости от используемого диапазона электромагнитного спектра: широковещательные (радио-) системы, микроволновые системы, системы инфракрасных волн, системы видимого света.

Из-за отражения, дифракции и рассеивания электромагнитных волн возникает многолучевое распространение одного и того же сигнала. Это приводит к межсимвольной интерференции и многолучевому замиранию.

Передача данных в диапазонах 900 МГц, 2,4 ГГц и 5 ГГц, которые получили название ISM-диапазонов, не требует лицензирования, если мощность передатчика не превышает 1 Вт.

Беспроводные двухточечные линии связи служат для создания радиорелейных линий, соединения зданий, а также пары компьютеров.

Беспроводные линии связи с одним источником и несколькими приемниками строятся на основе базовой станции. Такие линии используются в мобильных сотовых сетях, а также в системах фиксированного доступа.

Топология с несколькими источниками и несколькими приемниками характерна для беспроводных локальных сетей.

В системах спутниковой связи используются три группы спутников: геостационарные, среднеорбитальные и низкоорбитальные.

Для кодирования дискретной информации в беспроводных системах прибегают к манипуляции (FSK и PSK), модуляции с несколькими несущими частотами (OFDM) и методам расширения спектра (FHSS и DSSS).

В методах расширения спектра для представления информации используется широкий диапазон частот, это уменьшает влияние на сигналы узкополосных шумов.

На основе методов FHSS и DSSS можно мультиплексировать несколько каналов в одном диапазоне частот. Такая техника мультиплексирования называется множественным доступом с кодовым разделением (CDMA).

Вопросы и задания

1. Назовите основные области применения беспроводных линий связи.
2. В чем достоинства и недостатки беспроводной передачи информации по сравнению с проводной?
3. Антенна какого типа является направленной? Варианты ответов:
4. а) параболическая; б) изотропная.
5. За счет чего радиоволны с частотами от 2 до 30 МГц могут распространяться на сотни километров?
6. Какой спектр волн используется для спутниковой связи?
7. Какие атмосферные явления мешают распространению микроволн?

8. Что из ниже перечисленного используется для ненаправленного распространения инфракрасных волн:
   а) лазерные диоды;
   б) система линз;
   в) отражение от потолка;
   г) тепловые антенны.
9. Какие препятствия вызывают дифракцию? Варианты ответов:
   а) непроницаемые препятствия, размер которых соизмерим с длиной волны;
   б) непроницаемые препятствия, размер которых намного больше длины волны;
   в) непроницаемые препятствия, размер которых намного меньше длины волны.
10. В каких случаях применяются эллиптические орбиты телекоммуникационных спутников?
11. Какими недостатками обладает геостационарный спутник? Варианты ответов:
   а) велики задержки сигнала;
   б) велико затухание сигнала, что приводит к необходимости использования антенн большого диаметра;
   в) мало покрытие территории;
   г) плохая связь в районах, близких к северному и южному полюсам.
12. При соблюдении какого условия технология FHSS является высокоскоростной?
13. Какое свойство последовательности Баркера определяет возможность ее использования в технологии DSSS?
14. Назовите основное свойство расширяющих последовательностей, используемых в технологии CDMA.
15. Можно ли в качестве расширяющих последовательностей узлов сети, поддерживающих множественный доступ с кодовым разделением на основе технологии DSSS, использовать значения 100...0, 0100...0, 0010...0, 00010...0 и т. д.?
16. Предложите 11-битную расширяющую последовательность, отличную от последовательности Баркера, которая, как и последовательность Баркера, позволяет надежно определять начало передачи очередного бита исходной информации.

ГЛАВА 11 Первичные сети

Первичные сети предназначены для создания коммутируемой инфраструктуры, с помощью которой можно достаточно быстро и гибко организовать постоянный канал с двухточечной топологией между двумя пользовательскими устройствами, подключенными к такой сети. В первичных сетях применяется техника коммутации каналов. На основе каналов, образованных первичными сетями, работают наложенные компьютерные или телефонные сети. Каналы, предоставляемые первичными сетями своим пользователям, отличаются высокой пропускной способностью — обычно от 2 Мбит/с до 10 Гбит/с.

Существует несколько поколений технологий первичных сетей:

- плезиохронная цифровая иерархия (Plesiochronous Digital Hierarchy, PDH);
- синхронная цифровая иерархия (Synchronous Digital Hierarchy, SDH) — этой технологии в Америке соответствует стандарт SONET;
- уплотненное волновое мультиплексирование (Dense Wave Division Multiplexing, DWDM);
- оптические транспортные сети (Optical Transport Network, OTN) — данная технология определяет способы передачи данных по волновым каналам DWDM.

В технологиях PDH, SDH и OTN для разделения высокоскоростного канала применяется временно́е мультиплексирование (TDM), а данные передаются в цифровой форме. Каждая из них поддерживает иерархию скоростей, так что пользователь может выбрать подходящую ему скорость для каналов, с помощью которых он будет строить наложенную сеть.

Технологии OTN и SDH обеспечивают более высокие скорости, чем технология PDH, так что при построении крупной первичной сети ее магистраль строится на технологии OTN или SDH, а сеть доступа — на технологии PDH.

Сети DWDM не являются собственно цифровыми сетями, так как предоставляют своим пользователям выделенную волну для передачи информации, которую те могут применять по своему усмотрению — модулировать или кодировать. Техника мультиплексирования DWDM существенно повысила пропускную способность современных телекоммуникационных сетей, так как она позволяет организовать в одном оптическом волокне несколько десятков волновых каналов, каждый из которых может переносить цифровую информацию. В начальный период развития технологии DWDM волновые каналы использовались в основном для передачи сигналов SDH, то есть мультиплексоры DWDM были одновременно и мультиплексорами SDH для каждого из своих волновых каналов.

Впоследствии для более эффективного использования волновых каналов DWDM была разработана технология OTN, которая позволяет передавать по волновым каналам сигналы любых технологий, включая SDH, Gigabit Ethernet и 10G Ethernet.

Сети PDH

Технология PDH была разработана в конце 60-х годов компанией AT&T для решения проблемы связи крупных коммутаторов телефонных сетей между собой. Линии связи FDM, применяемые ранее для решения этой задачи, исчерпали свои возможности в плане организации высокоскоростной многоканальной связи по одному кабелю. В технологии FDM для одновременной передачи данных 12 абонентских каналов использовалась витая пара, а для повышения скорости связи приходилось прокладывать кабели с большим количеством пар проводов или более дорогие коаксиальные кабели.

Иерархия скоростей

Начало технологии PDH было положено разработкой мультиплексора **T-1**, который позволял в цифровом виде мультиплексировать, передавать и коммутировать (на постоянной основе) голосовой трафик 24 абонентов. Так как абоненты по-прежнему пользовались обычными телефонными аппаратами, то есть передача голоса шла в аналоговой форме, то мультиплексоры T-1 сами осуществляли оцифровывание голоса с частотой 8000 Гц и кодировали голос методом импульсно-кодовой модуляции. В результате каждый абонентский канал образовывал цифровой поток данных 64 Кбит/с, а мультиплексор T-1 обеспечивал передачу 1,544 Мбит/с.

В качестве средств мультиплексирования при соединении крупных телефонных станций каналы T-1 были слишком медленны и негибки, поэтому была реализована идея образования каналов с *иерархией скоростей*. Четыре канала типа T-1 объединили в канал следующего уровня цифровой иерархии — T-2, передающий данные со скоростью 6,312 Мбит/с. Канал T-3, образованный путем объединения семи каналов T-2, имеет скорость 44,736 Мбит/с. Канал T-4 объединяет 6 каналов T-3, в результате его скорость равна 274 Мбит/с. Описанная технология получила название **системы T-каналов**.

С середины 70-х годов выделенные каналы, построенные на основе систем T-каналов, стали сдаваться телефонными компаниями в аренду на коммерческих условиях, перестав быть внутренней технологией этих компаний. Системы T-каналов позволяют передавать не только голос, но и любые данные, представленные в цифровой форме: компьютерные данные, телевизионное изображение, факсы и т. п.

Технология систем T-каналов была стандартизована Американским национальным институтом стандартов (ANSI), а позже — международной организацией ITU-T. При стандартизации она получила название плезиохронной цифровой иерархии (PDH). В результате внесенных ITU-T изменений возникла несовместимость американской и международной версий стандарта PDH. Аналогом систем T-каналов в международном стандарте являются каналы типа **E-1**, **E-2** и **E-3** с отличающимися скоростями — соответственно 2,048 Мбит/с, 8,488 Мбит/с и 34,368 Мбит/с. Американская версия сегодня помимо США распространена также в Канаде и Японии (с некоторыми различиями), в Европе же применяется международный стандарт ITU-T.

Несмотря на различия, в американской и международной версиях технологии цифровой иерархии принято использовать одни и те же обозначения для иерархии скоростей — DSn (Digital Signal n). В табл. 11.1 приводятся значения для всех введенных стандартами уровней скоростей обеих технологий.

Таблица 11.1. Иерархия цифровых скоростей

Америка				ITU-T (Европа)		
Обозначение скорости	**Количество голосовых каналов**	**Количество каналов предыдущего уровня**	**Скорость, Мбит/с**	**Количество голосовых каналов**	**Количество каналов предыдущего уровня**	**Скорость, Мбит/с**
DS-0	1	1	64 Кбит/с	1	1	64 Кбит/с
DS-1	24	24	1,544	30	30	2,048
DS-2	96	4	6,312	120	4	8,488
DS-3	672	7	44,736	480	4	34,368
DS-4	4032	6	274,176	1920	4	139,264

На практике в основном используются каналы T-1/E-1 и T-3/E-3.

Методы мультиплексирования

Мультиплексор T-1 обеспечивает передачу данных 24-х абонентов со скоростью 1,544 Мбит/с в кадре, имеющем достаточно простой формат. В этом кадре последовательно передается по одному байту каждого абонента, а после 24 байт вставляется один *бит синхронизации*. Первоначально устройства T-1 (которые дали имя всей технологии, работающей на скорости 1,544 Мбит/с) функционировали только на внутренних тактовых генераторах, и каждый кадр с помощью битов синхронизации мог передаваться асинхронно. Аппаратура T-1 (а также более скоростная аппаратура T-2 и T-3) за долгие годы существования претерпела значительные изменения.

Сегодня мультиплексоры и коммутаторы первичной сети работают на централизованной тактовой частоте, распределяемой из одной или нескольких точек сети.

Однако принцип формирования кадра остался, поэтому биты синхронизации в кадре по-прежнему присутствуют. Суммарная скорость пользовательских каналов составляет $24 \times 64 = 1{,}536$ Мбит/с, а еще 8 Кбит/с добавляют биты синхронизации, итого получается 1,544 Мбит/с.

Теперь рассмотрим еще одну особенность формата кадра T-1. В аппаратуре T-1 восьмой бит каждого байта в кадре имеет назначение, зависящее от типа передаваемых данных и поколения аппаратуры. При передаче *голоса* с помощью этого бита переносится служебная информация, к которой относятся номер вызываемого абонента и другие сведения, необходимые для установления соединения между абонентами сети. Протокол, обеспечивающий такое соединение, называется в телефонии **сигнальным протоколом**. Поэтому реальная скорость передачи пользовательских данных в этом случае составляет не 64, а 56 Кбит/с. Техника применения восьмого бита для служебных целей получила название **«кражи» бита**.

При передаче компьютерных данных канал T-1 предоставляет для пользовательских данных только 23 канала, а 24-й канал отводится для служебных целей, в основном – для восстановления искаженных кадров. Компьютерные данные передаются со скоростью 64 Кбит/с, так как восьмой бит не «крадется».

При одновременной передаче как голосовых, так и компьютерных данных используются все 24 канала, причем и компьютерные, и голосовые данные передаются со скоростью 56 Кбит/с

При мультиплексировании 4-х каналов T-1 в один канал T-2 между кадрами DS-1 по-прежнему передается один бит синхронизации, а кадры DS-2 (которые состоят из 4-х последовательных кадров DS-1) разделяются 12 служебными битами, предназначенными не только для разделения кадров, но и для их синхронизации. Соответственно, кадры DS-3 состоят из 7 кадров DS-2, разделенных служебными битами.

Версия технологии PDH, описанная в международных стандартах G.700–G.706 ITU-T, как уже отмечалось, имеет отличия от американской технологии систем T-каналов. В частности, в ней не используется схема «кражи бита». При переходе к следующему уровню иерархии коэффициент кратности скорости имеет постоянное значение 4. Вместо восьмого бита в канале E-1 на служебные цели отводятся 2 байта из 32, а именно нулевой (для целей синхронизации приемника и передатчика) и шестнадцатый (в нем передается служебная сигнальная информация). Для голосовых или компьютерных данных остается 30 каналов со скоростью передачи 64 Кбит/с каждый.

При мультиплексировании нескольких пользовательских потоков в мультиплексорах PDH применяется техника, называемая **бит-стаффингом**. К этой технике прибегают, когда скорость пользовательского потока оказывается несколько меньше, чем скорость объединенного потока — подобные проблемы могут возникать в сети, состоящей из большого количества мультиплексоров, несмотря на все усилия по централизованной синхронизации узлов сети (в природе нет ничего идеального, в том числе идеально синхронных узлов сети). В результате мультиплексор PDH периодически сталкивается с ситуацией, когда ему «не хватает» бита для представления в объединенном потоке того или иного пользовательского потока. В этом случае мультиплексор просто вставляет в объединенный поток бит-вставку и отмечает этот факт в служебных битах объединенного кадра. При демультиплексировании объединенного потока бит-вставка удаляется из пользовательского потока, который возвращается в исходное состояние. Техника бит-стаффинга применяется как в международной, так и в американской версиях PDH.

Отсутствие полной синхронности потоков данных при объединении низкоскоростных каналов в высокоскоростные и дало название технологии PDH («плезиохронный» означает «почти синхронный»).

Пользователь может арендовать несколько каналов 64 Кбит/с (56 Кбит/с) в канале T-1/E-1. Такой канал называется «дробным» каналом T-1/E-1. В этом случае пользователю отводится несколько тайм-слотов работы мультиплексора.

Физический уровень технологии PDH поддерживает различные виды кабелей: витую пару, коаксиальный кабель, волоконно-оптический кабель. Основным вариантом абонентского доступа к каналам T-1/E-1 является кабель из двух витых пар с разъемами RJ-48. Две пары требуются для организации дуплексного режима передачи данных со скоростью 1,544/2,048 Мбит/с. Для представления сигналов используются:

- в каналах T-1 — биполярный потенциальный код B8ZS;
- в каналах E-1 — биполярный потенциальный код HDB3.

Для усиления сигнала на линиях T-1 через каждые 1800 м (одна миля) устанавливаются регенераторы и аппаратура контроля линии.

Коаксиальный кабель благодаря своей широкой полосе пропускания поддерживает один канал T-2/E-2 или 4 канала T-1/E-1. Для работы каналов T-3/E-3 обычно используется либо коаксиальный кабель, либо волоконно-оптический кабель, либо каналы СВЧ.

Физический уровень международного варианта технологии определяется стандартом G.703. Название этого стандарта служит также для обозначения типа интерфейса маршрутизатора или моста, подключаемого к каналу E-1. Американский вариант названия интерфейса — T-1.

Синхронизация сетей PDH

В случае небольшой сети PDH, например сети города, синхронизация всех устройств сети из одной точки представляется достаточно простым делом. Однако для более крупных сетей, например сетей масштаба страны, которые состоят из некоторого количества региональных сетей, синхронизация всех устройств сети представляет собой проблему.

Общий подход к решению этой проблемы описан в стандарте ITU-T G.810. Он заключается в организации в сети иерархии эталонных источников синхросигналов, а также системы распределения синхросигналов по всем узлам сети (рис. 11.1).

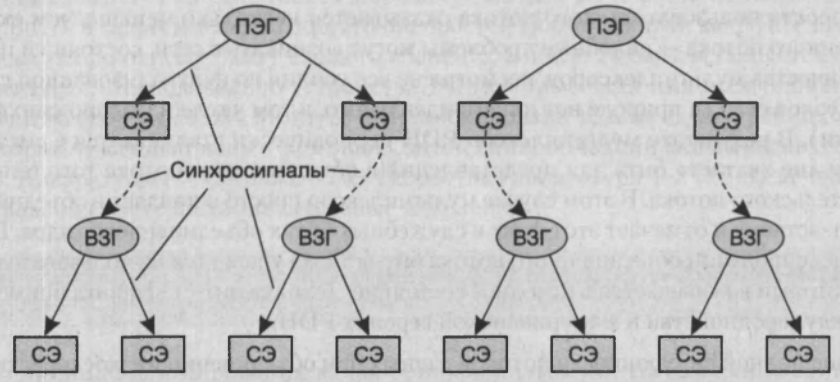


Рис. 11.1. Организация распределения синхросигналов по узлам сети PDH

Каждая крупная сеть должна иметь, по крайней мере, один **первичный эталонный генератор** (ПЭГ) синхросигналов (в англоязычном варианте — Primary Reference Clock, PRC). Это очень точный источник синхросигналов, способный вырабатывать синхросигналы с относительной точностью частоты не хуже 10–11 (такую точность требуют стандарты ITU-T G.811 и ANSI T1.101, в последнем для описания точности ПЭГ применяется название **Stratum 1**). На практике в качестве ПЭГ используют либо автономные атомные (водородные или цезиевые) часы, либо часы, синхронизирующиеся от спутниковых систем точного мирового времени, таких как GPS или ГЛОНАСС. Обычно точность ПЭГ достигает 10–13.

Стандартным синхросигналом является сигнал тактовой частоты уровня DS1, то есть частоты 2048 кГц для международного варианта стандартов PDH и 1544 кГц для американского варианта этих стандартов.

Синхросигналы от ПЭГ непосредственно поступают на специально отведенные для этой цели синхровходы магистральных устройств сети PDH. В том случае, если это составная

сеть, то каждая крупная сеть, входящая в состав составной сети (например, региональная сеть, входящая в состав национальной сети), имеет свой ПЭГ.

Для синхронизации немагистральных узлов используется **вторичный задающий генератор (ВЗГ)** синхросигналов, который в варианте ITU-T называют Secondary Reference Clock (SRC), а в варианте ANSI — генератор уровня **Stratum 2**. ВЗГ работает в режиме принудительной синхронизации, являясь ведомым таймером в паре ПЭГ-ВЗГ. Обычно ВЗГ получает синхросигналы от некоторого ПЗГ через промежуточные магистральные узлы сети, при этом для передачи синхросигналов используются биты служебных байтов кадра, например нулевого байта кадра E-1 в международном варианте PDH.

Точность ВЗГ меньше, чем точность ПЭГ: ITU-T в стандарте G.812 определяет ее как «не хуже 10^{-9}», а точность генераторов Stratum 2 должна быть не «хуже $1{,}6 \times 10^{-8}$».

Иерархия эталонных генераторов может быть продолжена, если это необходимо, при этом точность каждого более низкого уровня естественно понижается. Генераторы нижних уровней, начиная от ВЗГ, могут использовать для выработки своих синхросигналов несколько эталонных генераторов более высокого уровня, но при этом в каждый момент времени один из них должен быть основным, а остальные — резервными; такое построение системы синхронизации обеспечивает ее отказоустойчивость. Однако в этом случае нужно приоритезировать сигналы генераторов более высоких уровней. Кроме того, при построении системы синхронизации нужно гарантировать отсутствие петель синхронизации.

Методы синхронизации цифровых сетей, кратко описанные в этом разделе, применимы не только к сетям PDH, но и к другим сетям, работающих на основе синхронного TDM-мультиплексирования, например к сетям SDH, а также к сетям цифровых телефонных коммутаторов.

Ограничения технологии PDH

Как американский, так и международный варианты технологии PDH обладают недостатками, основным из которых является сложность и неэффективность операций мультиплексирования и демультиплексирования пользовательских данных. Применение техники бит-стаффинга для выравнивания скоростей потоков приводит к тому, что для извлечения пользовательских данных из объединенного канала необходимо полностью (!) демультиплексировать кадры объединенного канала.

Например, чтобы получить данные одного абонентского канала 64 Кбит/с из кадров канала T-3, требуется произвести демультиплексирование этих кадров до уровня кадров T-2, затем – до уровня кадров T-1, а в конце концов демультиплексировать и сами кадры T-1.

Если сеть PDH используется только в качестве транзитной магистрали между двумя крупными узлами, то операции мультиплексирования и демультиплексирования выполняются исключительно в конечных узлах, и проблем не возникает. Но если необходимо выделить один или несколько абонентских каналов в промежуточном узле сети PDH, то эта задача простого решения не имеет. Как вариант предлагается установка двух мультиплексоров уровня T3/E3 и выше в каждом узле сети (рис. 11.2). Первый призван обеспечить полное демультиплексирование потока и отвод части низкоскоростных каналов абонентам, второй — опять собрать в выходной высокоскоростной поток оставшиеся каналы вместе с вновь вводимыми. При этом количество работающего оборудования удваивается.

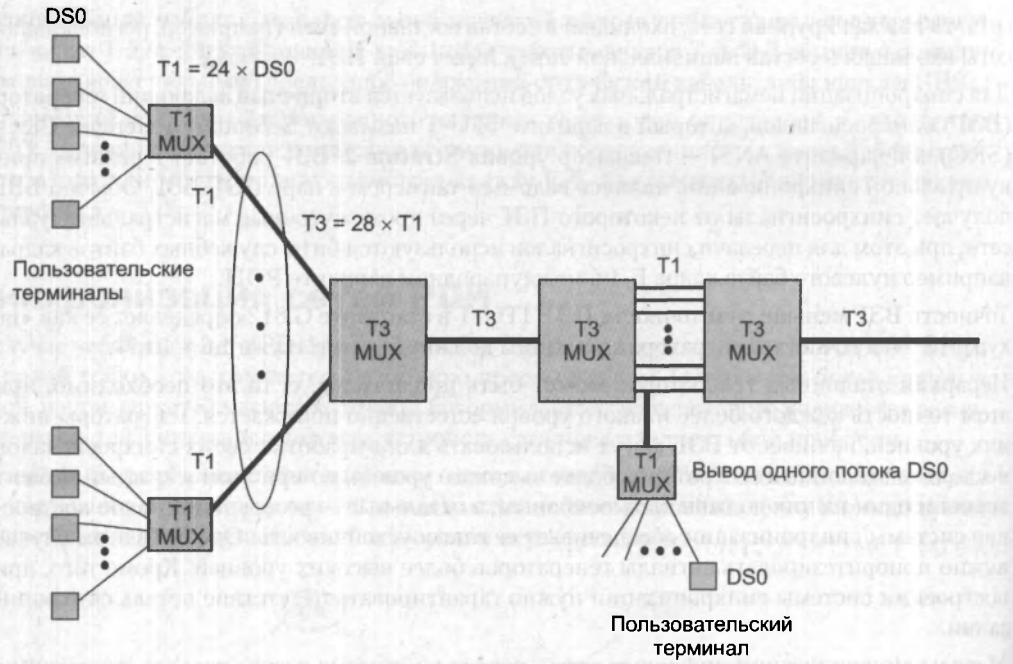


Рис. 11.2. Выделение низкоскоростного канала путем полного демультиплексирования

Другой вариант — «обратная доставка». В промежуточном узле, где нужно выделить и отвести абонентский поток, устанавливается единственный высокоскоростной мультиплексор, который просто передает данные транзитом дальше по сети без их демультиплексирования. Эту операцию выполняет только мультиплексор конечного узла, после чего данные соответствующего абонента возвращаются по отдельной линии связи в промежуточный узел. Естественно, такие сложные взаимоотношения коммутаторов усложняют работу сети, требуют ее тонкого конфигурирования, что ведет к большому объему ручной работы и ошибкам.

К тому же в технологии PDH не предусмотрены встроенные средства обеспечения отказоустойчивости и администрирования сети.

Наконец, недостатком PDH являются слишком низкие по современным понятиям скорости передачи данных. Волоконно-оптические кабели позволяют передавать данные со скоростями в несколько гигабит в секунду по одному волокну, что обеспечивает консолидацию в одном кабеле десятков тысяч пользовательских каналов, но эту возможность технология PDH не реализует — ее иерархия скоростей заканчивается уровнем 139 Мбит/с.

Сети SONET/SDH

Характерные для технологии PDH недостатки были учтены и преодолены разработчиками технологии **синхронных оптических сетей** (Synchronous Optical NET, SONET), первый вариант стандарта которой появился в 1984 г. Затем она была стандартизована комитетом Т-1 института ANSI. Международная стандартизация технологии проходила под эгидой

Европейского института телекоммуникационных стандартов (European Telecommunications Standards Institute, ETSI) и сектором телекоммуникационной стандартизации союза ITU (ITU Telecommunication Standardization Sector, ITU-T) совместно с ANSI и ведущими телекоммуникационными компаниями Америки, Европы и Японии. Основной целью разработчиков международного стандарта было создание технологии, способной передавать трафик всех существующих цифровых каналов уровня PDH (как американских T1–T3, так и европейских E1–E4) по высокоскоростной магистральной сети на базе волоконно-оптических кабелей и обеспечить иерархию скоростей, продолжающую иерархию технологии PDH до скорости в несколько гигабит в секунду.

В результате длительной работы ITU-T и ETSI удалось подготовить международный стандарт **SDH** (Synchronous Digital Hierarchy — синхронная цифровая иерархия). Кроме того, стандарт SONET был доработан так, чтобы аппаратура и сети SDH и SONET являлись совместимыми и могли мультиплексировать входные потоки практически любого стандарта PDH — и американского, и европейского.

Иерархия скоростей и методы мультиплексирования

Поддерживаемая технологией SONET/SDH иерархия скоростей представлена в табл. 11.2.

Таблица 11.2. Иерархия скоростей SONET/SDH

SDH	SONET	Скорость
	STS-1, OC-1	51,84 Мбит/с
STM-1	STS-3, OC-3	155,520 Мбит/с
STM-3	OC-9	466,560 Мбит/с
STM-4	OC-12	622,080 Мбит/с
STM-6	OC-18	933,120 Мбит/с
STM-8	OC-24	1,244 Гбит/с
STM-12	OC-36	1,866 Гбит/с
STM-16	OC-48	2,488 Гбит/с
STM-64	OC-192	9,953 Гбит/с
STM-256	OC-768	39,81 Гбит/с

В стандарте SDH все уровни скоростей (и, соответственно, форматы кадров для этих уровней) имеют общее название STM-N (Synchronous Transport Module level N — синхронный транспортный модуль уровня N). В технологии SONET существует два обозначения для уровней скоростей: название STS-N (Synchronous Transport Signal level N — синхронный транспортный сигнал уровня N) употребляется в случае передачи данных электрическим сигналом, а название OC-N (Optical Carrier level N — оптоволоконная линия связи уровня N) используют в случае передачи данных по волоконно-оптическому кабелю. Далее для упрощения изложения мы сосредоточимся на технологии SDH.

Кадры STM-N имеют достаточно сложную структуру, позволяющую агрегировать в общий магистральный поток потоки SDH и PDH различных скоростей, а также выполнять операции ввода-вывода без полного демультиплексирования магистрального потока.

Операции мультиплексирования и ввода-вывода выполняются при помощи **виртуальных контейнеров** (Virtual Container, VC), в которых блоки данных PDH можно транспортировать через сеть SDH. Помимо блоков данных PDH в виртуальный контейнер помещается еще некоторая служебная информация, в частности **заголовок пути** (Path OverHead, POH) контейнера, в котором размещается статистическая информация о процессе прохождении контейнера вдоль пути от его начальной до конечной точки (сообщения об ошибках), а также другие служебные данные, например индикатор установления соединения между конечными точками. В результате размер виртуального контейнера оказывается больше, чем соответствующая нагрузка в виде блоков данных PDH, которую он переносит. Например, виртуальный контейнер VC-12 помимо 32 байт данных потока E-1 содержит еще 3 байта служебной информации.

В технологии SDH определено несколько типов виртуальных контейнеров (рис. 11.3), предназначенных для транспортировки основных типов блоков данных PDH: VC-11 (1,5 Мбит/с), VC-12 (2 Мбиг/с), VC-2 (6 Мбит/с), VC3 (34/45 Мбит/с) и VC-4 (140 Мбит/с).

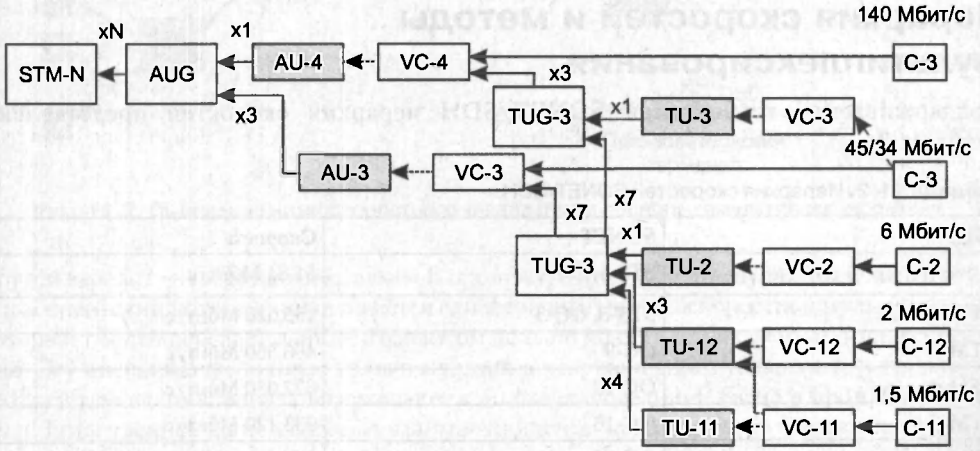


Рис. 11.3. Схема мультиплексирования данных в SDH

Виртуальные контейнеры являются *единицей коммутации* мультиплексоров SDH. В каждом мультиплексоре существует **таблица соединений** (называемая также **таблицей кросс-соединений**), в которой указано, например, что контейнер VC-12 порта P1 соединен с контейнером VC12 порта P5, а контейнер VC3 порта P8 — с контейнером VC3 порта P9. Таблицу соединений формирует администратор сети с помощью системы управления или управляющего терминала на каждом мультиплексоре так, чтобы обеспечить сквозной путь между конечными точками сети, к которым подключено пользовательское оборудование.

Чтобы совместить в рамках одной сети механизмы синхронной передачи кадров (STM-N) и асинхронный характер переносимых этими кадрами пользовательских данных PDH, в технологии SDH применяются **указатели**. Концепция указателей — ключевая в технологии SDH, она заменяет принятое в PDH выравнивание скоростей асинхронных источников посредством дополнительных битов. Указатель определяет текущее положение виртуального контейнера в агрегированной структуре более высокого уровня, каковой является **трибутарный блок** (Tributary Unit, TU) либо **административный блок** (Administrative Unit, AU). Собственно, основное отличие этих блоков от виртуального контейнера заключается в наличии дополнительного поля указателя. С помощью этого указателя виртуальный

контейнер может «смещаться» в определенных пределах внутри своего трибутарного или административного блока, если скорость пользовательского потока несколько отличается от скорости кадра SDH, куда этот поток мультиплексируется.

Именно благодаря системе указателей мультиплексор находит положение пользовательских данных в синхронном потоке байтов кадров STM-N и «на лету» извлекает их оттуда, чего механизм мультиплексирования, применяемый в PDH, делать не позволяет.

Трибутарные блоки объединяются в группы, а те, в свою очередь, входят в административные блоки. Группа административных блоков (Administrative Unit Group, AUG) в количестве *N* и образует полезную нагрузку кадра STM-N. Помимо этого в кадре имеется заголовок с общей для всех блоков AU служебной информацией. На каждом шаге преобразования к предыдущим данным добавляется несколько служебных байтов: они помогают распознать структуру блока или группы блоков и затем определить с помощью указателей начало пользовательских данных.

На рис. 11.3 структурные единицы кадра SDH, содержащие указатели, заштрихованы, а связь между контейнерами и блоками, допускающая сдвиг данных по фазе, показана пунктиром.

Схема мультиплексирования SDH предоставляет разнообразные возможности по объединению пользовательских потоков PDH. Например, для кадра STM-1 можно реализовать такие варианты:

- 1 поток E-4;
- 63 потока E-1;
- 1 поток E-3 и 42 потока E-1.

Другие варианты читатель может предложить сам.

Типы оборудования

Основным элементом сети SDH является **мультиплексор** (рис. 11.4). Обычно он оснащен некоторым количеством портов PDH и SDH: например, портами PDH на 2 и 34/45 Мбит/с и портами SDH STM-1 на 155 Мбит/с и STM-4 на 622 Мбит/с. Порты мультиплексора SDH делятся на агрегатные и трибутарные.

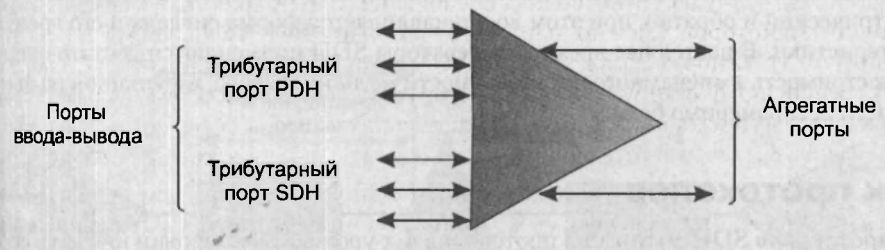


Рис. 11.4. Мультиплексор SDH

Трибутарные порты часто называют также портами ввода-вывода, а **агрегатные** — линейными портами. Эта терминология отражает типовые топологии сетей SDH, где имеется ярко выраженная магистраль в виде цепи или кольца, по которой передаются потоки данных, поступающие от пользователей сети через порты ввода-вывода (трибутарные порты), то есть втекающие в агрегированный поток («tributary» дословно означает «приток»).

Мультиплексоры SDH обычно разделяют на два типа, разница между которыми определяется положением мультиплексора в сети SDH (рис. 11.5).

Терминальный мультиплексор (Terminal Multiplexer, TM) *завершает* агрегатный канал, мультиплексируя в нем большое количество трибутарных каналов, поэтому он оснащен одним агрегатным и большим числом трибутарных портов.

Мультиплексор ввода-вывода (Add-Drop Multiplexer, ADM) занимает промежуточное положение на магистрали (в кольце, цепи или смешанной топологии). Он имеет два агрегатных порта, транзитом передавая агрегатный поток данных. С помощью небольшого количества трибутарных портов такой мультиплексор вводит в агрегатный поток или выводит из агрегатного потока данные трибутарных каналов.

Рис. 11.5. Типы мультиплексоров SDH

Иногда также выделяют мультиплексоры, которые выполняют операции коммутации над произвольными виртуальными контейнерами — так называемые **цифровые кросс-коннекторы** (Digital Cross-Connect, DXC). В таких мультиплексорах не делается различий между агрегатными и трибутарными портами, так как они предназначены для работы в ячеистой топологии, где выделить агрегатные потоки невозможно.

Помимо мультиплексоров, в состав сети SDH могут входить **регенераторы сигналов**, необходимые для преодоления ограничений по расстоянию между мультиплексорами. Эти ограничения зависят от мощности оптических передатчиков, чувствительности приемников и затухания волоконно-оптического кабеля. Регенератор преобразует оптический сигнал в электрический и обратно, при этом восстанавливается форма сигнала и его временные характеристики. В настоящее время регенераторы SDH применяются достаточно редко, так как стоимость их ненамного ниже стоимости мультиплексора, а функциональные возможности несоизмеримо беднее.

Стек протоколов

Стек протоколов SDH состоит из протоколов 4-х уровней. Эти уровни никак не соотносятся с уровнями модели OSI, для которой вся сеть SDH представляется как оборудование физического уровня.

Фотонный уровень имеет дело с кодированием битов информации путем модуляции света. Для кодирования оптического сигнала применяется потенциальный код NRZ, обладающий свойствами самосинхронизации.

Уровень секции поддерживает физическую целостность сети. **Регенераторной секцией** в технологии SDH называется каждый непрерывный отрезок волоконно-оптического кабеля, который соединяет между собой такие, например, пары устройств SONET/SDH, как мультиплексор и регенератор, регенератор и регенератор, но не два мультиплексора. Компоненты регенераторной секции поддерживают протокол, который имеет дело с определенной частью заголовка кадра, называемой **заголовком регенераторной секции** (Regenerator Section OverHead, RSOH), и который на основе служебной информации может проводить тестирование секции и выполнять операции административного контроля.

Уровень линии отвечает за передачу данных *по линии* между двумя мультиплексорами сети, поэтому линию также часто называют **мультиплексной секцией**. Протокол этого уровня работает с кадрами уровней STS-N для выполнения различных операций мультиплексирования и демультиплексирования, а также вставки и удаления пользовательских данных. Кроме того, протокол линии ответственен за реконфигурирование линии в случае отказа какого-либо ее элемента — оптического волокна, порта или соседнего мультиплексора. Служебная информация мультиплексной секции располагается в части заголовка кадра, называемой **заголовком мультиплексной секции** (Multiplex Section OverHead, MSOH).

Уровень тракта отвечает за доставку данных между двумя конечными пользователями сети. Тракт — это составное виртуальное соединение между пользователями. Протокол тракта должен принять данные, поступающие в пользовательском формате, например формате T-1, и преобразовать их в синхронные кадры STM-N.

На рис. 11.6 показано распределение протоколов SDH по типам оборудования SDH.

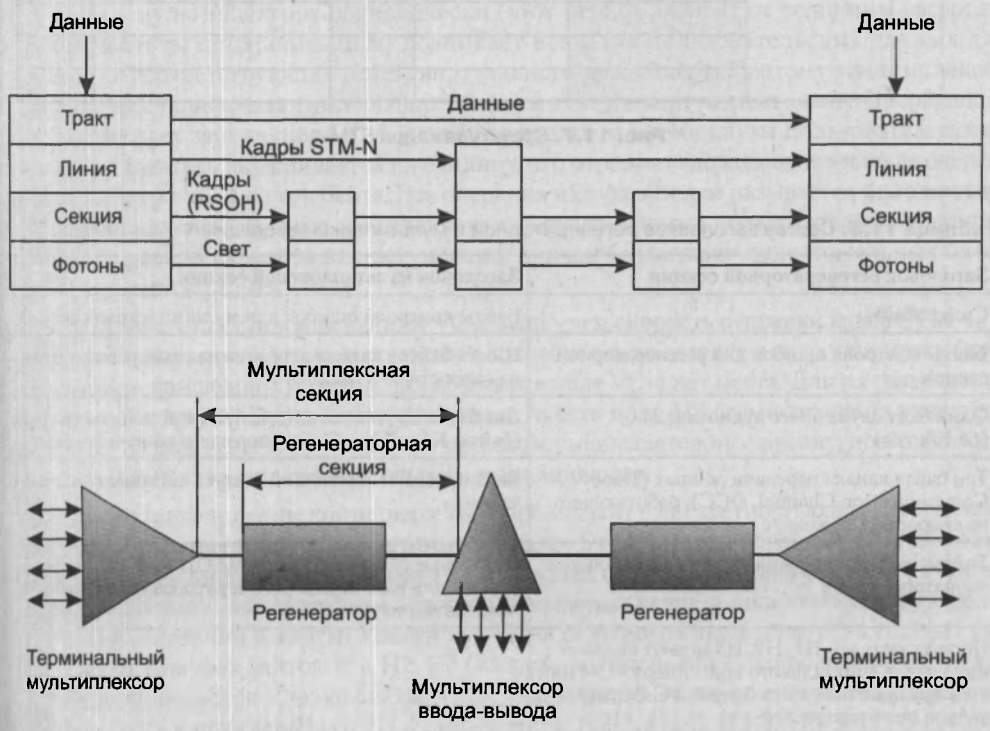


Рис. 11.6. Стек протоколов технологии SDH

Кадры STM-N

Основные элементы кадра STM-1 показаны на рис. 11.7, а в табл. 11.3 приведена структура заголовков регенераторной и мультиплексной секций.

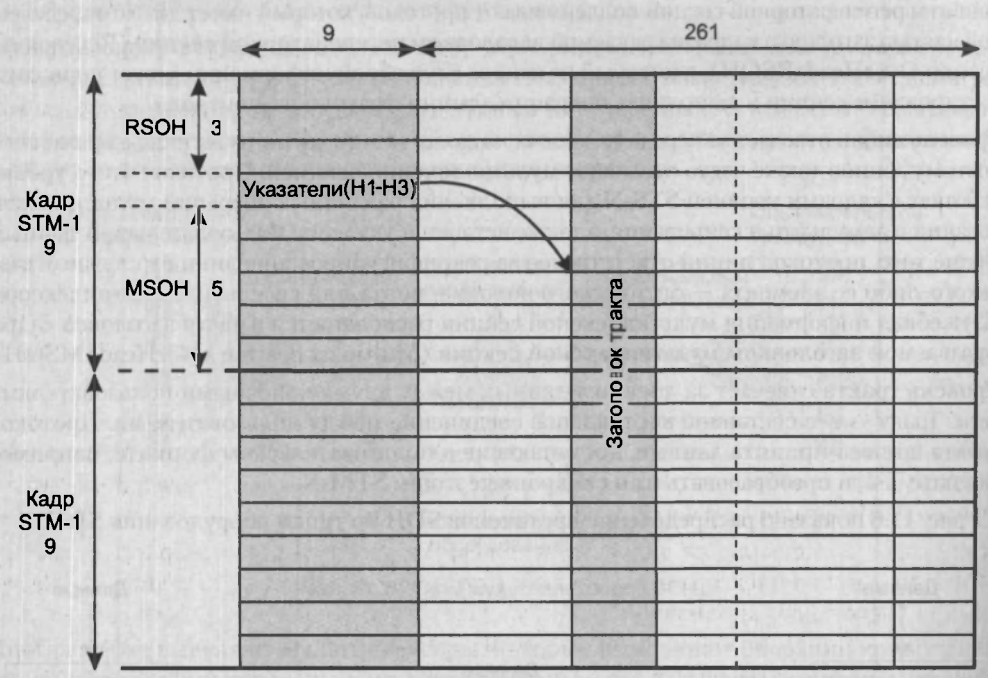


Рис. 11.7. Структура кадра STM-1

Таблица 11.3. Состав заголовков регенераторной и мультиплексной секций

Заголовок регенераторной секции	Заголовок мультиплексной секции
Синхробайты	Байты контроля ошибок для мультиплексной секции
Байты контроля ошибок для регенераторной секции	Шесть байтов канала передачи данных, работающего на скорости 576 Кбит/с
Один байт служебного аудиоканала (64 Кбит/с)	Два байта протокола автоматической защиты трафика (байты K1 и K2), обеспечивающего живучесть сети
Три байта канала передачи данных (Data Communication Channel, DCC), работающего на скорости 192 Кбит/с	Байт передачи сообщений статуса системы синхронизации
Байты, зарезервированные для национальных операторов связи	Остальные байты заголовка MSOH либо зарезервированы для национальных операторов связи, либо не используются
Поля указателей H1, H2, H3 задают положение начала виртуального контейнера VC-4 или трех виртуальных контейнеров VC-3 относительно поля указателей	

Кадр обычно представляют в виде матрицы, состоящей из 270 столбцов и 9 строк. Первые 9 байт каждой строки отводятся под служебные данные заголовков, из последующих 261 байт 260 отводятся под полезную нагрузку (данные таких структур, как AUG, AU, TUG, TU и VC), а один байт каждой строки — под **заголовок тракта**, что позволяет контролировать соединение «из конца в конец».

Рассмотрим механизм работы указателя H1-H2-H3 на примере кадра STM-1, переносящего контейнер VC-4. Указатель занимает 9 байт четвертого ряда кадра, причем под каждое из полей H1, H2 и H3 в этом случае отводится по 3 байта. Разрешенные значения указателя находятся в диапазоне 0–782, причем указатель отмечает начало контейнера VC-4 в трехбайтовых единицах. Например, если указатель имеет значение 27, то первый байт VC-4 находится на расстоянии 27 × 3 = 81 байт от последнего байта поля указателей, то есть является 90-м байтом (нумерация начинается с единицы) в 4-й строке кадра STM-1. Фиксированное значение указателя позволяет учесть фазовый сдвиг между данным мультиплексором и источником данных, в качестве которого может выступать мультиплексор PDH, оборудование пользователя с интерфейсом PDH или другой мультиплексор SDH. В результате виртуальный контейнер передается в двух последовательных кадрах STM-1, как и показано на рис. 11.7.

Указатель может отрабатывать не только фиксированный фазовый сдвиг, но и рассогласование тактовой частоты мультиплексора с тактовой частотой устройства, от которого принимаются пользовательские данные. Для компенсации этого эффекта значение указателя периодически наращивается или уменьшается на единицу.

Если скорость поступления данных контейнера VC-4 меньше, чем скорость отправки STM-1, то у мультиплексора периодически (этот период зависит от величины рассогласования частоты синхронизации) возникает нехватка пользовательских данных для заполнения соответствующих полей виртуального контейнера. Поэтому мультиплексор вставляет три «холостых» (незначащих) байта в данные виртуального контейнера, после чего продолжает заполнение VC-4 «подоспевшими» за время паузы пользовательскими данными. Указатель наращивается на единицу, что отражает запаздывание начала очередного контейнера VC-4 на три байта. Эта операция над указателем называется **положительным выравниванием**. В итоге средняя скорость отправляемых пользовательских данных становится равной скорости их поступления, причем без вставки дополнительных битов в стиле технологии PDH.

Если же скорость поступления данных VC-4 выше, чем скорость отправки кадра STM-1, то у мультиплексора периодически возникает потребность во вставке в кадр «лишних» (преждевременно пришедших) байтов, для которых в поле VC-4 нет места. Для их размещения используются три младших байта указателя, то есть поле H3 (само значение указателя умещается в поля H1 и H2). Указатель при этом уменьшается на единицу, поэтому такая операция носит название **отрицательного выравнивания**.

Тот факт, что выравнивание контейнера VC-4 происходит с дискретностью в три байта, объясняется достаточно просто. Дело в том, что в кадре STM-1 может переноситься либо один контейнер VC-4, либо три контейнера VC-3. Каждый из контейнеров VC-3 имеет в общем случае независимое значение фазы относительно начала кадра, а также собственную величину рассогласования частоты. Указатель VC-3 в отличие от указателя VC-4 состоит уже не из девяти, а из трех байтов: H1, H2, H3 (каждое из этих полей — однобайтовое). Эти три указателя помещаются в те же байты, что и указатель VC-4, но по схеме с чередованием байтов, то есть в порядке H1-1, H1-2, H1-3, H2-1, H2-2, H2-3, H3-1, H3-2, H3-3 (второй

индекс идентифицирует определенный контейнер VC-3). Значения указателей VC-3 интерпретируются в байтах, а не трехбайтовых единицах. При отрицательном выравнивании контейнера VC-3 лишний байт помещается в соответствующий байт НЗ-1, НЗ-2 или НЗ-3 – в зависимости от того, над каким из контейнеров VC-3 проводится операция.

Вот мы и дошли до размера смещения для контейнеров VC4 — этот размер был выбран для унификации этих операций над контейнерами любого типа, размещаемыми непосредственно в AUG кадра STM-1. Выравнивание контейнеров более низкого уровня всегда происходит с шагом в один байт.

При объединении блоков TU и AU в группы в соответствии с описанной схемой (см. рис. 11.7) выполняется их последовательное побайтное расслоение, так что период следования пользовательских данных в кадре STM-N совпадает с периодом их следования в трибутарных портах. Это исключает необходимость в их временной буферизации, поэтому говорят, что *мультиплексоры SDH передают данные в реальном масштабе времени*.

Упомянутая ранее техника *прямой коррекции ошибок* (FEC) была стандартизована в технологии SDH гораздо позже принятия основного ядра стандартов SDH. Напомним, что эта техника основана на применении самокорректирующих кодов, позволяющих исправлять искажения битов данных «на лету», то есть не прибегая к их повторной передаче, а используя избыточную часть кода. Такая техника может существенно повысить эффективную скорость передачи данных при наличии помех или сбоев в работе приемопередатчиков. Обычно к прямой коррекции ошибок мультиплексоры SDH прибегают на скоростях 2,5 Гбит/с и выше.

Типовые топологии

В сетях SDH применяются различные топологии связей. Наиболее часто используются кольца и линейные цепи мультиплексоров, также находит все большее применение ячеистая топология, близкая к полносвязной.

Кольцо SDH строится из мультиплексоров ввода-вывода, имеющих, по крайней мере, по два агрегатных порта (рис. 11.8, *а*). Пользовательские потоки вводятся в кольцо и выводятся из кольца через трибутарные порты, образуя двухточечные соединения (на рисунке показаны в качестве примера два таких соединения). Кольцо является классической регулярной топологией, обладающей потенциальной отказоустойчивостью — при однократном обрыве кабеля или выходе из строя мультиплексора соединение сохранится, если его направить по кольцу в противоположном направлении. Кольцо обычно строится на основе кабеля с двумя оптическими волокнами, но иногда для повышения надежности и пропускной способности применяют четыре волокна.

Цепь (рис. 11.8, *б*) — это линейная последовательность мультиплексоров, из которых два оконечных играют роль терминальных мультиплексоров, остальные — мультиплексоров ввода-вывода. Обычно сеть с топологией цепи применяется в тех случаях, когда узлы имеют соответствующее географическое расположение, например вдоль магистрали железной дороги или трубопровода. Правда, в таких случаях может применяться и **плоское кольцо** (рис. 11.8, *в*), обеспечивающее более высокий уровень отказоустойчивости за счет двух дополнительных волокон в магистральном кабеле и по одному дополнительному агрегатному порту у терминальных мультиплексоров.

Эти базовые топологии могут комбинироваться при построении сложной и разветвленной сети SDH, образуя участки с радиально-кольцевой топологией, соединениями «кольцо-кольцо» и т. п. Наиболее общим случаем является **ячеистая топология** (рис. 11.8, *г*), при

которой мультиплексоры соединяются друг с другом большим количеством связей, за счет чего сеть можно достичь очень высокой степени производительности и надежности.

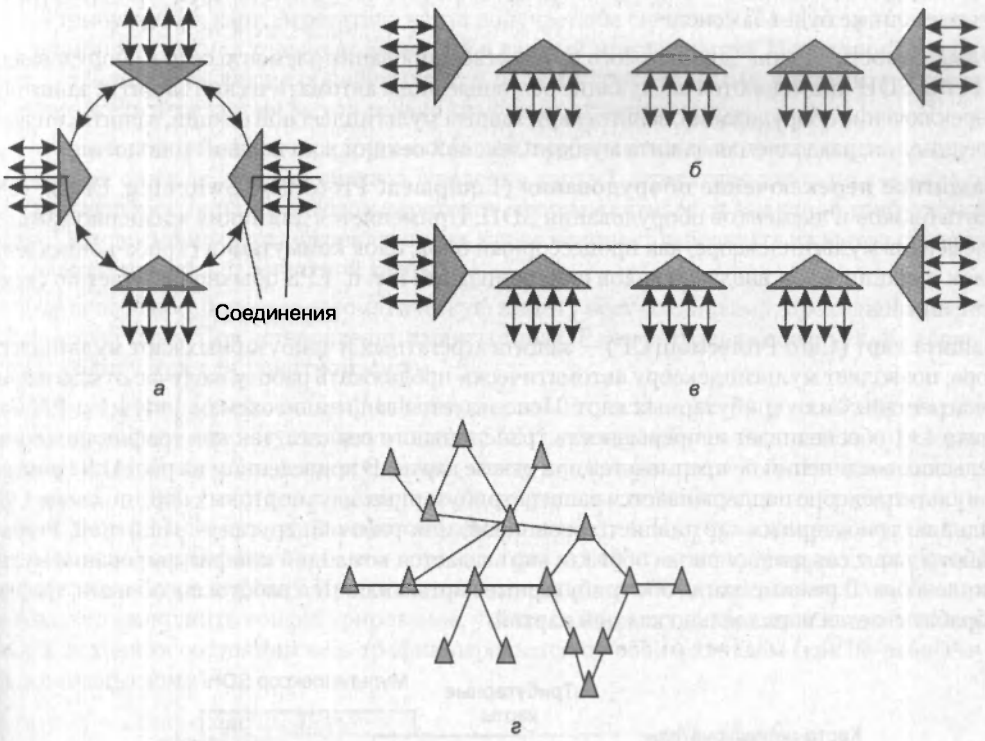


Рис. 11.8. Типовые топологии

Методы обеспечения живучести сети

Одной из сильных сторон первичных сетей SDH является разнообразный набор средств отказоустойчивости, который позволяет сети быстро (за десятки миллисекунд) восстановить работоспособность в случае отказа какого-либо элемента сети — линии связи, порта или карты мультиплексора, мультиплексора в целом.

В SDH в качестве общего названия механизмов отказоустойчивости используется термин **автоматическое защитное переключение** (Automatic Protection Switching, APS), отражающий факт перехода (переключения) на резервный путь или резервный элемент мультиплексора при отказе основного. Сети, поддерживающие такой механизм, в стандартах SDH названы **самовосстанавливающимися**.

В сетях SDH применяются три схемы защиты.

- **Защита 1+1** означает, что резервный элемент выполняет ту же работу, что и основной. Например, при защите трибутарной карты по схеме 1+1 трафик проходит как через рабочую карту (резервируемую), так и через защитную (резервную).
- **Защита 1:1** подразумевает, что защитный элемент в нормальном режиме не выполняет функции защищаемого элемента, а переключается на них только в случае отказа.

- **Защита 1:N** предусматривает выделение одного защитного элемента на *N* защищаемых. При отказе одного из защищаемых элементов его функции начинает выполнять защитный, при этом остальные элементы остаются без защиты — до тех пор, пока отказавший элемент не будет заменен.

В зависимости от типа защищаемого путем резервирования элемента сети в оборудовании и сетях SDH применяются следующие основные виды автоматической защиты: защитное переключение оборудования, защита карт, защита мультиплексной секции, защита сетевого соединения, разделяемая защита мультиплексной секции в кольцевой топологии.

Защитное переключение оборудования (Equipment Protection Switching, EPS) — защита блоков и элементов оборудования SDH. Применяется для таких жизненно важных элементов мультиплексора, как процессорный блок, блок коммутации (кросс-коннектор), блок питания, блок ввода сигналов синхронизации и т. п. EPS обычно работает по схеме 1+1 или 1:1.

Защита карт (Card Protection, CP) — защита агрегатных и трибутарных карт мультиплексора; позволяет мультиплексору автоматически продолжать работу в случае отказа одной из агрегатных или трибутарных карт. Используется защита по схемам 1+1, 1:1 и 1:*N*. Защита 1+1 обеспечивает непрерывность транспортного сервиса, так как трафик пользовательских соединений не прерывается при отказе карты. В приведенном на рис. 11.9 примере в мультиплексоре поддерживается защита трибутарных двухпортовых карт по схеме 1+1. Одна из трибутарных карт является основной, или рабочей, другая — защитной. Режим работы пары связанных таким образом карт задается командой конфигурирования мультиплексора. В режиме, когда обе трибутарные карты являются работоспособными, трафик обрабатывается параллельно каждой картой.

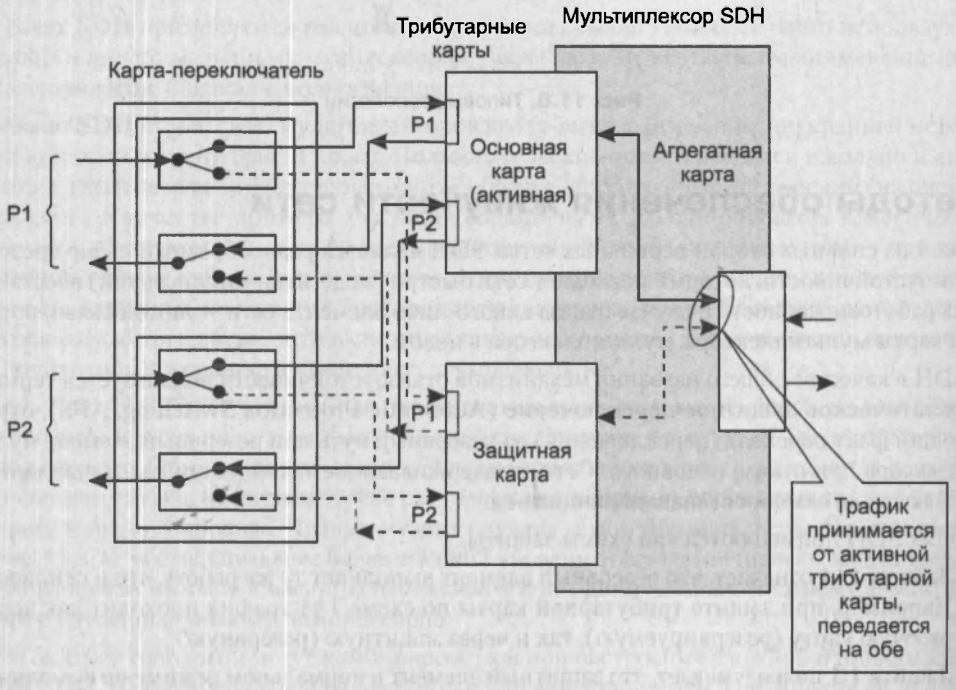


Рис. 11.9. Защита карт по схеме 1+1

Для переключения трафика между трибутарными картами используется дополнительная карта-переключатель. Входящий трафик каждого порта поступает на входной мост карты-переключателя, который разветвляет трафик и передает его на входы соответствующих портов трибутарных карт. Агрегатная карта получает оба сигнала STM-N от трибутарных карт и выбирает сигнал только от активной в данный момент карты. Исходящий трафик от агрегатной карты также обрабатывается обеими трибутарными картами, но карта-переключатель передает на выход только трафик от активной карты.

При отказе основной карты или другом событии, требующем перехода на защитную карту (деградация сигнала, ошибка сигнала, удаление карты), агрегатная карта по команде от блока управления мультиплексором переходит на прием сигнала от защитной трибутарной карты. Одновременно карта-переключатель также начинает передавать на выход сигналы выходящего трафика от защитной карты.

Данный способ обеспечивает автоматическую защиту всех соединений, проходящих через защищаемую карту. При установлении защиты типа CP конфигурация соединений рабочей карты дублируется для защитной карты.

Защита мультиплексной секции (Multiplex Section Protection, MSP), то есть участка сети между двумя смежными мультиплексорами SDH, действует более избирательно по сравнению с защитой карт. Защищается секция между двумя мультиплексорами, включающая два порта и линию связи (возможно, в свою очередь, включающую регенераторы, но не мультиплексоры). Обычно применяется схема защиты 1+1. При этом для рабочего канала (верхняя пара соединенных кабелем портов на рис. 11.10, *а*) конфигурируется защитный канал (нижняя пара портов). При установлении защиты MSP в каждом мультиплексоре необходимо выполнить конфигурирование, указав связь между рабочим и защитным портами. В исходном состоянии весь трафик передается по обоим каналам (как по рабочему, так и по защитному).

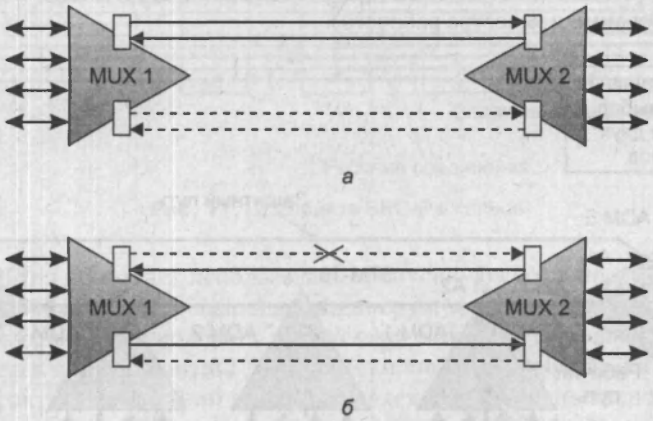


Рис. 11.10. Защита мультиплексной секции

Существует однонаправленная и двунаправленная защита MSP. При однонаправленной защите (именно этот случай показан на рисунке) решение о переключении принимает только один из мультиплексоров — тот, который является приемным для отказавшего канала. Этот мультиплексор после обнаружения отказа (отказ порта, ошибка сигнала, деградация сигнала и т. п.) переходит на прием по защитному каналу. При этом передача и прием ведутся по разным портам (рис. 11.20, *б*).

В случае двунаправленной защиты MSP при отказе рабочего канала в каком-либо направлении выполняется полное переключение на защитные порты мультиплексоров. Для уведомления передающего (по рабочему каналу) мультиплексора о необходимости переключения принимающий мультиплексор использует протокол, называемый протоколом «K-байт». Этот протокол указывает в двух байтах заголовка кадра STM-N статус рабочего и защитного каналов, а также детализирует информацию об отказе. Механизм MSP обеспечивает защиту всех соединений, проходящих через защищаемую мультиплексную секцию. Время переключения защиты MSP, согласно требованиям стандарта, не должно превышать 50 мс.

Защита сетевого соединения (Sub-Network Connection Protection, SNC-P), то есть защита пути (соединения) через сеть для определенного виртуального контейнера, обеспечивает переключение определенного пользовательского соединения на альтернативный путь при отказе основного пути. Объектом защиты SNC-P является трибутарный трафик, помещенный в виртуальный контейнер определенного типа (например, в VC-12, VC-3 или VC-4). Используется схема защиты 1+1.

Защита SNC-P конфигурируется в двух мультиплексорах: во входном, в котором трибутарный трафик, помещенный в виртуальный контейнер, разветвляется, и в выходном, в котором сходятся два альтернативных пути трафика. Пример защиты SNC-P показан на рис. 11.11. В мультиплексоре ADM 1 для виртуального контейнера VC-4 трибутарного порта T2 заданы два соединения: с одним из четырех контейнеров VC-4 агрегатного порта A1 и с одним из четырех контейнеров VC-4 агрегатного порта A2. Одно из соединений конфигурируется как рабочее, второе — как защитное, при этом трафик передается по обоим соединениям. Промежуточные (для данных соединений) мультиплексоры конфигурируются обычным образом. В выходном мультиплексоре контейнер VC-4 трибутарного порта T3 также соединяется с контейнерами — агрегатного порта A1 и агрегатного порта A2. Из двух поступающих на порт T3 потоков выбирается тот, качество которого выше (при равном нормальном качестве выбирается сигнал из агрегатного порта, указанного при конфигурировании в качестве рабочего).

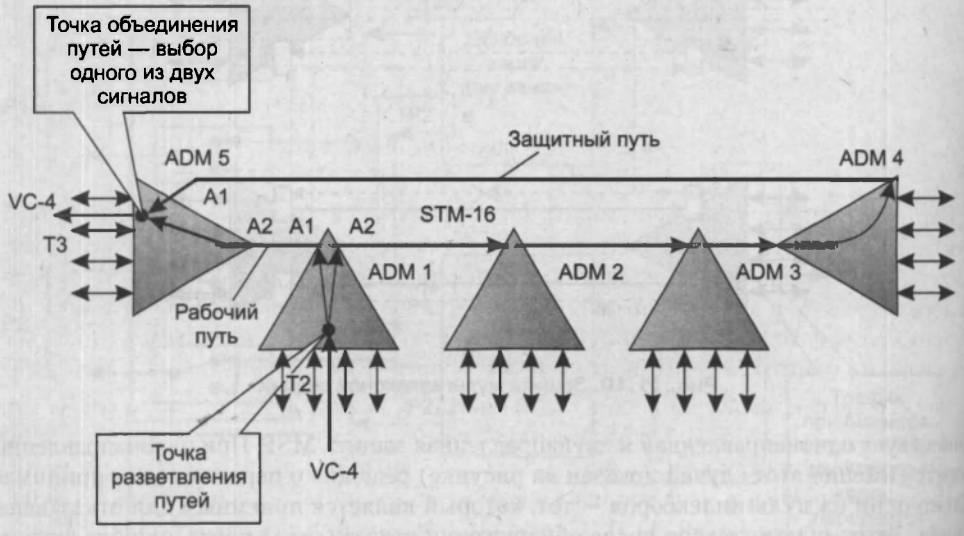


Рис. 11.11. Защита сетевого соединения

Защита SNC-P работает в любых топологиях сетей SDH, в которых имеются альтернативные пути следования трафика, то есть кольцевых и ячеистых.

Разделяемая защита мультиплексной секции в кольцевой топологии (Multiplex Section Shared Protection Ring, MS-SPRing) обеспечивает в некоторых случаях более экономичную защиту трафика в кольце. Хотя защита SNC-P вполне подходит для кольцевой топологии сети SDH, в некоторых случаях ее применение снижает полезную пропускную способность кольца, так как каждое соединение потребляет удвоенную полосу пропускания вдоль всего кольца. Так, в кольце STM-16 можно установить только 16 защищенных по типу SNC-P соединений VC-4 (рис. 11.12).

Рис. 11.12. Защита SNC-P в кольце

Защита MS-SPRing позволяет использовать пропускную способность кольца более эффективно, так как полоса пропускания не резервируется заранее для каждого соединения. Вместо этого резервируется половина пропускной способности кольца, но эта резервная полоса выделяется для соединений динамически, по мере необходимости, то есть после обнаружения факта отказа линии или мультиплексора. Степень экономии полосы при применении защиты MS-SPRing зависит от распределения трафика.

Если весь трафик сходится в один мультиплексор, то есть имеется звездообразное распределение, защита MS-SPRing экономии по сравнению с SNC-P вообще не дает. Пример такой ситуации представлен на рис. 11.13, *а*, где центром «тяготения» трафика является мультиплексор *A*, а в кольце установлены те же 16 защищенных соединений, что и в примере защиты SNC-P на рис. 11.12. Для защиты соединений резервируется 8 из 16 виртуальных контейнеров агрегатного потока STM-16.

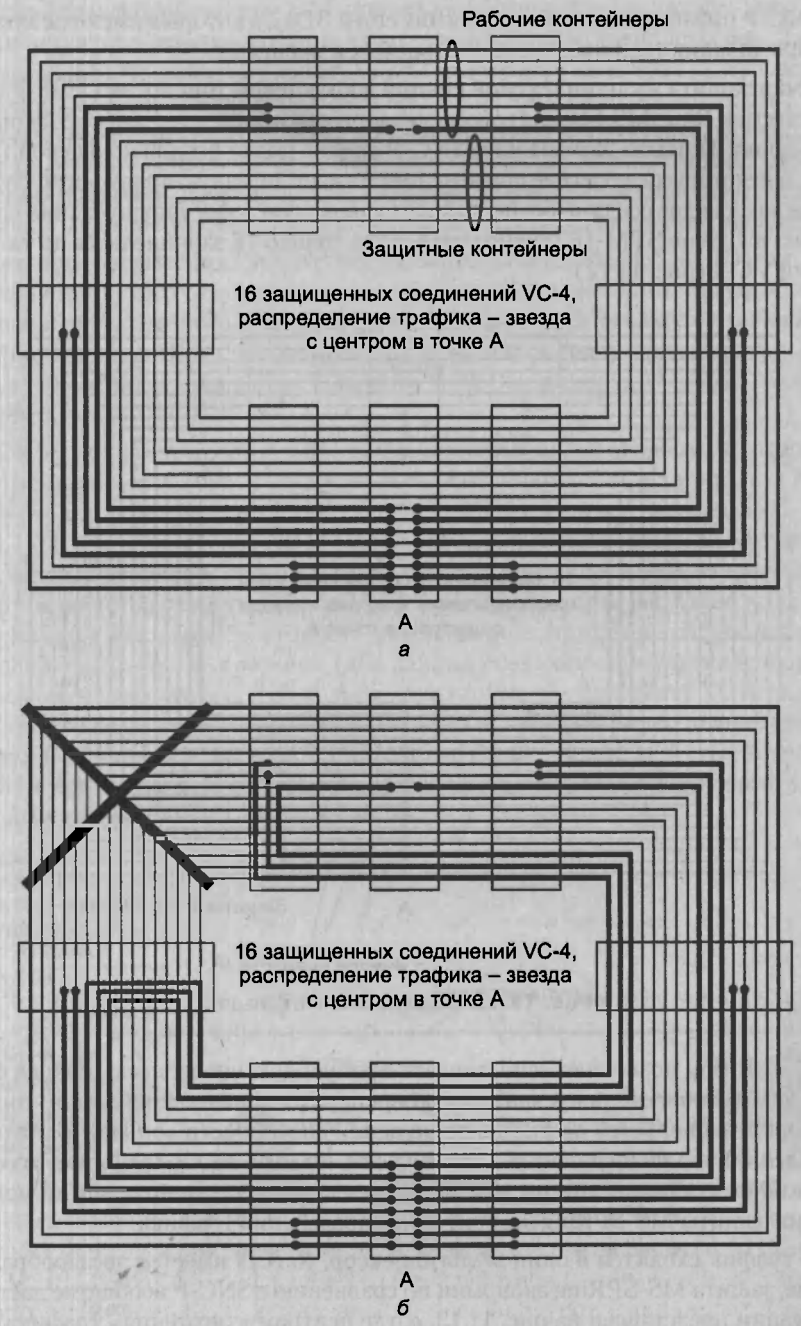


Рис. 11.13. Защита с разделением кольца

При возникновении неисправности, например обрыве линии, как это показано на рис. 11.13 *б*), трафик в мультиплексорах, между которыми нарушилась связь, «разво-

рачивается» в обратном направлении. Для этого используются резервные виртуальные контейнеры агрегатных портов, с которыми соединяются виртуальные контейнеры пострадавших соединений. В то же время соединения, на которые отказ не повлиял, работают в прежнем режиме, не подключая резервные контейнеры. Для уведомления мультиплексоров о реконфигурировании кольца служит уже упоминавшийся протокол «K-байт». Время переключения на защитные соединения MS-SPRing составляет около 50 мс. При смешанном распределении трафика экономия полосы в кольце MS-SPRing может быть еще более значительной.

Новое поколение протоколов SDH

Изначально технология SDH была ориентирована на передачу элементарных потоков голосового трафика, отсюда и ее ориентация на мультиплексирование пользовательских потоков со скоростями, кратными 64 Кбит/с, и применение коэффициента кратности 4 для иерархии скоростей.

Однако популярность Интернета изменила ситуацию в телекоммуникационном мире, и сегодня объемы компьютерного трафика в первичных сетях превосходят объемы голосового трафика. В условиях доминирования Ethernet как технологии канального уровня почти весь компьютерный трафик, поступающий на входы мультиплексоров первичных сетей, представляет собой кадры Ethernet, а значит, представлен иерархией скоростей 10-100-1000-10 000 Мбит/с. Пользовательские потоки с такими скоростями не очень эффективно укладываются в виртуальные контейнеры SDH, рассчитанные на решение других задач.

Для исправления ситуации организация ITU-T разработала несколько стандартов, которые составляют так называемую технологию SDH нового поколения (SDH Next Generation, или SDH NG). Эти стандарты делают технологию SDH более дружественной к компьютерным данным.

Стандарты SDH нового поколения описывают три новых механизма:

- виртуальная конкатенация (VCAT);
- схема динамического изменения пропускной способности линии (LCAS);
- общая процедура инкапсуляции (кадрирования) данных (GFP).

Виртуальная конкатенация (Virtual Concatenation, VCAT) контейнеров позволяет более эффективно использовать емкость виртуальных контейнеров SDH при передаче трафика Ethernet.

У механизма виртуальной конкатенации существует предшественник — механизм **смежной конкатенации**. Этот механизм был разработан для более эффективной передачи трафика сетей ATM; он позволяет объединить несколько контейнеров VC-4 со скоростью 140 Мбит/с в один контейнер с более высокой скоростью передачи данных. Коэффициент кратности объединения контейнеров VC-4 в механизме смежной конкатенации может быть равен 4, 16, 64 или 256, что позволяет использовать для передачи нескольких объединенных (конкатенированных) контейнеров VC-4 в кадрах STM-4, STM-16, STM-64 или STM-256. Объединенный контейнер рассматривается как единица коммутации всеми мультиплексорами сети, он имеет только один указатель, так как отдельные виртуальные контейнеры внутри объединенного контейнера заполняются данными одного потока и не могут «плавать» друг относительно друга. При смежной конкатенации объединенный контейнер обозначается как VC-4-4/16/64/256c.

Виртуальная конкатенация расширяет возможности смежной конкатенации за счет использования при объединении виртуальных контейнеров не только типа VC-4, но и других типов: VC-3 (34 Мбит/с), VC-12 (2 Мбит/с), VC-11 (1,5 Мбит/с) и VC-2 (6 Мбит/с). При этом объединяться могут лишь виртуальные контейнеры одного типа, например только VC-3 или только VC-12.

Кроме того, коэффициент кратности при объединении может быть любым от 1 до максимального числа, определяемого емкостью кадра STM-N, применяемого для передачи объединенного контейнера. При виртуальной конкатенации объединенный контейнер обозначается как VC-N-Mv, где N — тип виртуального контейнера, а M — кратность его использования, например, VC-3-21v.

Название «виртуальная конкатенация» отражает тот факт, что только конечные мультиплексоры (то есть тот мультиплексор, который формирует объединенный контейнер из пользовательских потоков, и тот мультиплексор, который его демультиплексирует в пользовательские потоки) должны понимать, что это — конкатенированный контейнер. Все промежуточные мультиплексоры сети SDH рассматривают составляющие виртуальные контейнеры как независимые и могут передавать их к конечному мультиплексору по разным маршрутам. Конечный мультиплексор выдерживает некоторый тайм-аут перед демультиплексированием пользовательских потоков, что может быть необходимо для прибытия всех составляющих контейнеров в том случае, когда они передаются по разным маршрутам.

Виртуальная конкатенация позволяет намного эффективнее расходовать пропускную способность сети SDH при передаче трафика Ethernet. Например, чтобы передавать один поток Fast Ethernet 100 Мбит/с, в сети STM-16 можно применить виртуальную конкатенацию VC-12-46v, которая обеспечивает пропускную способность для пользовательских данных 100,096 Мбит/с (то есть дает почти 100-процентную загрузку объединенного контейнера), а оставшиеся 206 контейнеров VC-12 (кадр STM-4 вмещает 63 × 4 = 252 контейнера VC-12) задействовать как для передачи других потоков Fast Ethernet, так и для передачи голосового трафика.

Схема динамического изменения пропускной способности линии (Link Capacity Adjustment Scheme, LCAS) является дополнением к механизму виртуальной конкатенации. Эта схема позволяет исходному мультиплексору, то есть тому, который формирует объединенный контейнер, динамически изменять его емкость, присоединяя к нему или отсоединяя от него виртуальные контейнеры. Для того чтобы добиться нужного эффекта, исходный мультиплексор посылает конечному мультиплексору специальное служебное сообщение, уведомляющее об изменении состава объединенного контейнера.

Общая процедура инкапсуляции данных (Generic Framing Procedure, GFP) предназначена для упаковки кадров различных протоколов компьютерных сетей в кадр единого формата и передачи его по сети SDH. Такая процедура полезна, так как она решает несколько задач, общих при передаче данных компьютерных сетей через сети SDH. В эти задачи входят выравнивание скорости компьютерного протокола со скоростью виртуального контейнера SDH, используемого для передачи компьютерных данных, а также распознавание начала кадра.

- *Выравнивание скорости компьютерного протокола и скорости виртуального контейнера SDH, используемого для передачи компьютерных данных.* Например, если мы применяем объединенный контейнер VC-12-46v для передачи кадров Fast Ethernet, то нужно выровнять скорости 100 и 100,096 Мбит/с. Процедура GFP поддерживает два режима

работы: **GFP-F** (кадровый режим, или Frame Mode) и **GFP-T** (прозрачный режим, или Transparent Mode). В режиме GFP-F проблема выравнивания скоростей решается обычным для компьютерных сетей способом — поступающий кадр полностью буферизуется, упаковывается в формат GFP, а затем со скоростью соединения SDH передается через сеть. Режим GFP-T предназначен для чувствительного к задержкам трафика, в этом режиме кадр полностью не буферизуется, а побитно по мере поступления передается в сеть SDH (предварительно снабженный служебными полями GFP). Для выравнивания скоростей в режиме GFP-T применяются специальные служебные «пустые» кадры GFP, которые посылаются в те моменты, когда рассогласование приводит к отсутствию пользовательских битов у исходного мультиплексора SDH.

- *Распознавание начала кадра.* Соединение SDH представляет для пользователя поток битов, разбитый на кадры SDH, начало которых никак не связано с началом кадра пользователя. Процедура GFP позволяет принимающему мультиплексору SDH распознать начало каждого пользовательского кадра, что необходимо для его извлечения из потока битов, проверки его корректности и передачи на выходной интерфейс в сеть пользователя. В процедуре GFP для распознавания начала кадра служит его собственный заголовок, который состоит из поля длины размером в два байта и поля контрольной суммы поля длины также размером в два байта. Для того чтобы «поймать» начало кадра, мультиплексор SDH последовательно смещается бит за битом по полученным данным, для каждого такого смещения вычисляет контрольную сумму для первых двух байтов данных, которые должны быть полем длины, и сравнивает вычисленное значение со значением, находящимся во вторых двух байтах данных. Если эти значения совпадают, мультиплексор считает, что данное смещение в полученных данных соответствует началу кадра — и с большой степенью вероятности так оно и есть. Если же значения не совпадают, это значит, что начало кадра не соответствует текущему смещению, тогда мультиплексор смещается на один бит дальше и повторяет свои вычисления. В конце концов, он доходит до положения, когда первый бит смещения действительно является первым битом поля длины кадра, при этом вычисляемая контрольная сумма совпадает с помещенной в кадр, и процесс распознавания заканчивается успешно. После этого мультиплексор долгое время находится в синхронизме с поступающими кадрами, то есть он постоянно с первого раза находит начало кадра — до тех пор, пока из-за каких-то помех не произойдет рассинхронизация и ему не придется методом последовательных смещений опять искать начало кадра.

Кроме описанных двух функций процедура GFP поддерживает еще ряд функций, полезных при передаче компьютерных данных по сетям SDH.

Сети DWDM

Технология **уплотненного волнового мультиплексирования** (Dense Wave Division Multiplexing, DWDM) предназначена для создания оптических магистралей нового поколения, работающих на мультигигабитных и терабитных скоростях. Такой революционный скачок производительности обеспечивает принципиально иной, нежели у SDH, метод мультиплексирования — информация в оптическом волокне передается одновременно большим количеством световых волн — **лямбд** — термин возник в связи с традиционным для физики обозначением длины волны λ.

Сети DWDM работают по принципу коммутации каналов, при этом каждая световая волна представляет собой отдельный *спектральный канал* и несет собственную информацию.

Оборудование DWDM не занимается непосредственно проблемами передачи данных на каждой волне, то есть способом кодирования информации и протоколом ее передачи. Его основными функциями являются операции *мультиплексирования* и *демультиплексирования*, а именно — объединение различных волн в одном световом пучке и выделение информации каждого спектрального канала из общего сигнала. Наиболее развитые устройства DWDM могут также *коммутировать* волны.

ВНИМАНИЕ

Технология DWDM является революционной не только потому, что в десятки раз повышает верхний предел скорости передачи данных по оптическому волокну, но и потому, что открывает новую эру в технике мультиплексирования и коммутации, выполняя эти операции над световыми сигналами без преобразования их в электрическую форму. Во всех других технологиях, в которых световые сигналы также используются для нередачи информации по оптическим волокнам, например SDH и Gigabit Ethernet, световые сигналы обязательно преобразуются в электрические и только потом их можно мультиплексировать и коммутировать.

Первым применением технологии DWDM были протяженные магистрали, предназначенные для связи двух сетей SDH. При такой простейшей двухточечной топологии способность устройств DWDM выполнять коммутацию волн является излишней, однако по мере развития технологии и усложнения топологий сетей DWDM эта функция становится востребованной.

Принципы работы

Сегодня оборудование DWDM позволяет передавать по одному оптическому волокну 32 и более волн разной длины в окне прозрачности 1550 нм, при этом каждая волна может переносить информацию со скоростью до 10 Гбит/с (при условии применения для передачи информации на каждой волне протоколов технологии STM или 10 Gigabit Ethernet). В настоящее время ведутся работы по повышению скорости передачи информации на одной длине волны до 40–80 Гбит/с.

У технологии DWDM имеется предшественница — технология **волнового мультиплексирования** (Wave Division Multiplexing, WDM), в которой используется четыре спектральных канала в окнах прозрачности 1310 нм и 1550 нм с разносом несущих в 800–400 ГГц. (Поскольку стандартной классификации WDM не существует, встречаются системы WDM и с другими характеристиками.)

Мультиплексирование DWDM называется «уплотненным» из-за того, что в нем используется существенно меньшее расстояние между длинами волн, чем в WDM. На сегодня рекомендацией G.692 сектора ITU-T определены два *частотных плана* (то есть набора частот, отстоящих друг от друга на некоторую постоянную величину):

- частотный план с шагом (разнесением частот между соседними каналами) **100 ГГц** ($\Delta\lambda \approx 0{,}8$ нм), в соответствии с которым для передачи данных применяется 41 волна в диапазоне от 1528,77 (196,1 ТГц) до 1560,61 нм (192,1 ТГц);
- частотный план с шагом **50 ГГц** ($\Delta\lambda \approx 0{,}4$ нм), позволяющий передавать в этом же диапазоне 81 длину волны.

Некоторыми компаниями выпускается также оборудование, называемое оборудованием **высокоуплотненного волнового мультиплексирования** (High-Dense WDM, HDWDM), способное работать с частотным планом с шагом **25 ГГц** (сегодня это чаще всего экспериментальные образцы, а не серийная продукция).

Реализация частотных планов с шагом 50 и 25 ГГц предъявляет гораздо более жесткие требования к оборудованию DWDM, особенно в том случае, если каждая волна переносит сигналы со скоростью модуляции 10 Гбит/с и выше (STM-64, 10GE или STM-256). Еще раз подчеркнем, что сама технология DWDM (как и WDM) не занимается непосредственно кодированием переносимой на каждой волне информации — это проблема более высокоуровневой технологии, которая пользуется предоставленной ей волной по своему усмотрению и может передавать на этой волне как дискретную, так и аналоговую информацию. Такими технологиями могут быть SDH или 10 Gigabit Ethernet. Теоретически зазоры между соседними волнами в 50 ГГц и даже 25 ГГц позволяют передавать данные со скоростями 10 Гбит/с, но при этом нужно обеспечить высокую точность частоты и минимально возможную ширину спектра несущей волны, а также снизить уровень шумов, чтобы минимизировать эффект перекрытия спектра (рис. 11.14).

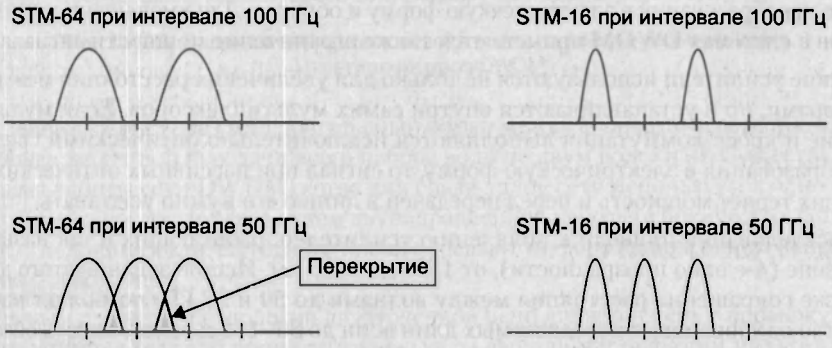


Рис. 11.14. Перекрытие спектра соседних волн для разных частотных планов и скоростей передачи данных

Волоконно-оптические усилители

Практический успех технологии DWDM, оборудование которой уже работает на магистралях многих ведущих мировых операторов связи, во многом определило появление волоконно-оптических усилителей. Эти оптические устройства непосредственно усиливают световые сигналы в диапазоне 1550 нм, исключая необходимость промежуточного преобразования их в электрическую форму, как это делают регенераторы, применяемые в сетях SDH. Системы электрической регенерации сигналов весьма дороги и, кроме того, зависят от протокола, так как они должны воспринимать определенный вид кодирования сигнала. Оптические усилители, «прозрачно» передающие информацию, позволяют наращивать скорость магистрали без необходимости модернизировать усилительные блоки.

Протяженность участка между оптическими усилителями может достигать 150 км и более, что обеспечивает экономичность создаваемых магистралей DWDM, в которых длина мультиплексной секции составляет на сегодня 600–3000 км при применении от 1 до 7 промежуточных оптических усилителей.

В рекомендации ITU-T G.692 определено три типа усилительных участков, то есть участков между двумя соседними мультиплексорами DWDM:

- **L** (Long) — участок состоит максимум из 8 пролетов волоконно-оптических линий связи и 7 оптических усилителей, максимальное расстояние между усилителями — до 80 км при общей максимальной протяженности участка 640 км;
- **V** (Very long) — участок состоит максимум из 5 пролетов волоконно-оптических линий связи и 4 оптических усилителей, максимальное расстояние между усилителями — до 120 км при общей максимальной протяженности участка 600 км;
- **U** (Ultra long) — участок без промежуточных усилителей длиной до 160 км.

Ограничения на количество пассивных участков и их длину связаны с деградацией оптического сигнала при оптическом усилении. Хотя оптический усилитель восстанавливает мощность сигнала, он не полностью компенсирует эффект хроматической дисперсии (то есть распространения волн разной длины с разной скоростью, из-за чего сигнал на приемном конце волокна «размазывается»), а также другие нелинейные эффекты. Поэтому для построения более протяженных магистралей необходимо между усилительными участками устанавливать мультиплексоры DWDM, выполняющие регенерацию сигнала путем его преобразования в электрическую форму и обратно. Для уменьшения нелинейных эффектов в системах DWDM применяется также ограничение мощности сигнала.

Оптические усилители используются не только для увеличения расстояния между мультиплексорами, но и устанавливаются внутри самих мультиплексоров. Если мультиплексирование и кросс-коммутация выполняются исключительно оптическими средствами без преобразования в электрическую форму, то сигнал при пассивных оптических преобразованиях теряет мощность и перед передачей в линию его нужно усиливать.

Новые исследования привели к появлению усилителей, работающих в так называемом L-диапазоне (4-е окно прозрачности), от 1570 до 1605 нм. Использование этого диапазона, а также сокращение расстояния между волнами до 50 и 25 ГГц позволяет нарастить количество одновременно передаваемых длин волн до 80–160 и более, то есть обеспечить передачу трафика со скоростями 800 Гбит/с–1,6 Тбит/с в одном направлении по одному оптическому волокну. С успехами DWDM связано еще одно перспективное технологическое направление — **полностью оптические сети**. В таких сетях все операции по мультиплексированию/демультиплексированию, вводу-выводу и кросс-коммутации (маршрутизации) пользовательской информации выполняются без преобразования сигнала из оптической формы в электрическую. Исключение преобразований в электрическую форму позволяет существенно удешевить сеть. Однако возможности оптических технологий пока еще недостаточны для создания масштабных полностью оптических сетей, поэтому их практическое применение ограничено фрагментами, между которыми выполняется электрическая регенерация сигнала.

Типовые топологии

Хронологически первой областью применения технологии DWDM (как и технологии SDH) стало создание сверхдальних высокоскоростных магистралей, имеющих топологию **двухточечной цепи** (рис. 11.15).

Для организации такой магистрали достаточно в ее конечных точках установить терминальные мультиплексоры DWDM, а в промежуточных точках — оптические усилители, если этого требует расстояние между конечными точками.

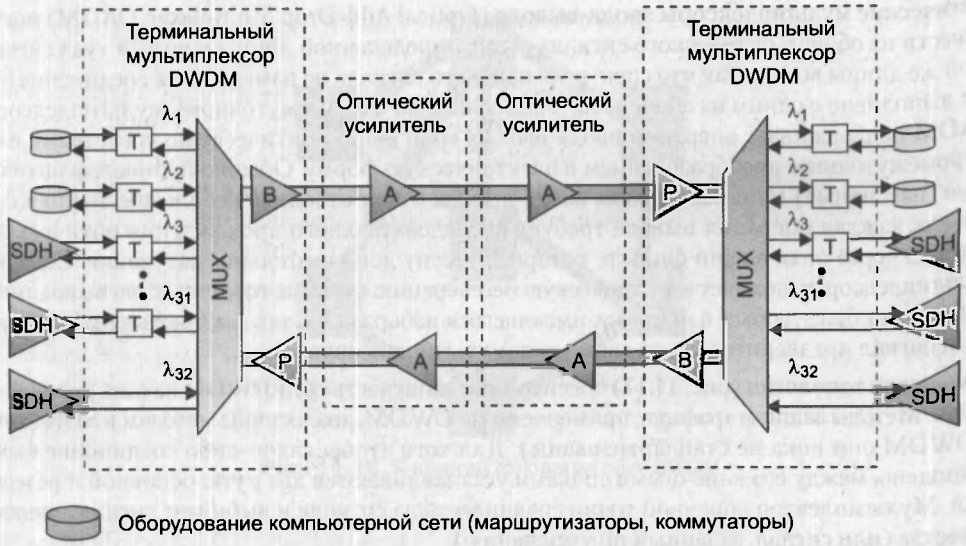


Рис. 11.15. Сверхдальняя двухточечная связь на основе терминальных мультиплексоров DWDM

В приведенной на рисунке схеме дуплексный обмен между абонентами сети происходит за счет однонаправленной передачи всего набора волн по двум волокнам. Существует и другой вариант работы сети DWDM, когда для связи узлов сети используется одно волокно. Дуплексный режим достигается путем двунаправленной передачи оптических сигналов по волокну — половина волн частотного плана передают информацию в одном направлении, половина — в обратном.

Естественным развитием топологии двухточечной цепи является **цепь с промежуточными подключениями**, в которой промежуточные узлы выполняют функции мультиплексоров ввода-вывода (рис. 11.16).

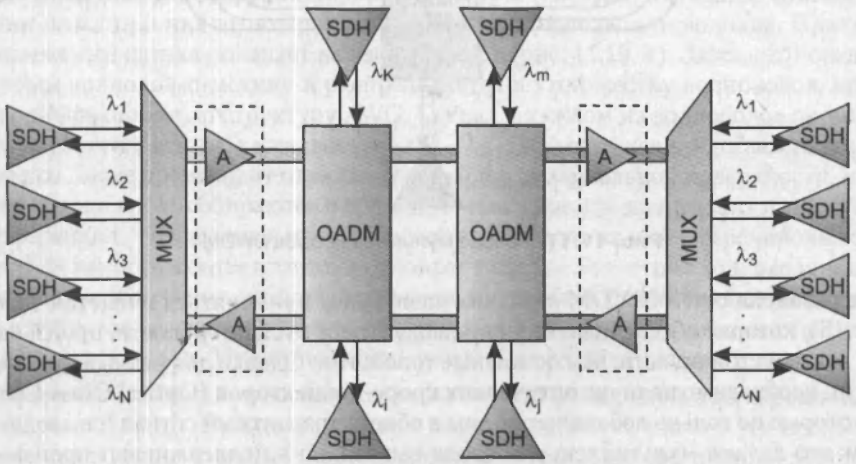


Рис. 11.16. Цепь DWDM с вводом-выводом в промежуточных узлах

Оптические мультиплексоры ввода-вывода (Optical Add-Drop Multiplexer, OADM) могут вывести из общего оптического сигнала волну определенной длины и ввести туда сигнал этой же длины волны, так что спектр транзитного сигнала не изменится, а соединение будет выполнено с одним из абонентов, подключенных к промежуточному мультиплексору. OADM поддерживает операции ввода-вывода волн сугубо оптическими средствами или с промежуточным преобразованием в электрическую форму. Обычно полностью оптические (пассивные) мультиплексоры ввода-вывода могут отводить небольшое число волн, так как каждая операция вывода требует последовательного прохождения оптического сигнала через оптический фильтр, который вносит дополнительное затухание. Если же мультиплексор выполняет электрическую регенерацию сигнала, то количество выводимых волн может быть любым в пределах имеющегося набора волн, так как транзитный оптический сигнал предварительно полностью демультиплексируется.

Кольцевая топология (рис. 11.17) обеспечивает живучесть сети DWDM за счет резервных путей. Методы защиты трафика, применяемые в DWDM, аналогичны методам в SDH (хотя в DWDM они пока не стандартизованы). Для того чтобы какое-либо соединение было защищено, между его конечными точками устанавливаются два пути: основной и резервный. Мультиплексор конечной точки сравнивает два сигнала и выбирает сигнал лучшего качества (или сигнал, заданный по умолчанию).

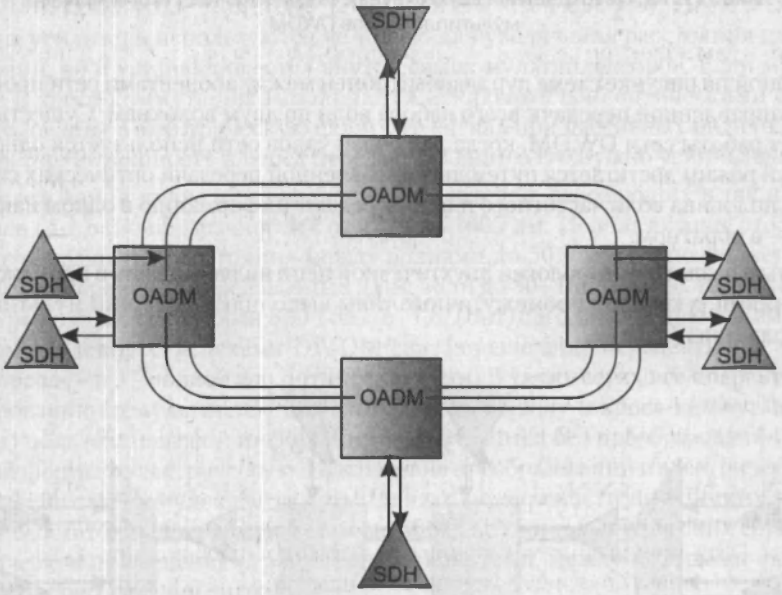


Рис. 11.17. Кольцо мультиплексоров DWDM

По мере развития сетей DWDM в них все чаще будет применяться **ячеистая топология** (рис. 11.18), которая обеспечивает лучшие показатели в плане гибкости, производительности и отказоустойчивости, чем остальные топологии. Однако для реализации ячеистой топологии необходимо наличие **оптических кросс-коннекторов** (Optical Cross-Connector, OXC), которые не только добавляют волны в общий транзитный сигнал и выводят их оттуда, как это делают мультиплексоры ввода-вывода, но и поддерживают произвольную коммутацию между оптическими сигналами, передаваемыми волнами разной длины.

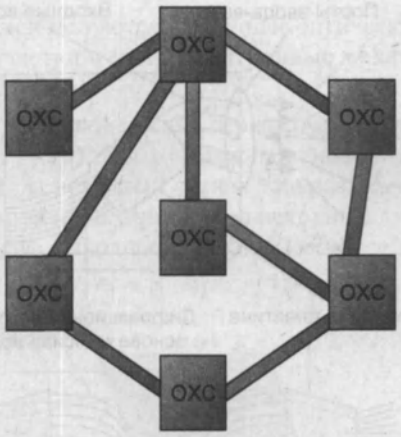


Рис. 11.18. Ячеистая топология сети DWDM

Оптические мультиплексоры ввода-вывода

Оптический мультиплексор выполняет операции смешения нескольких длин волн в общий сигнал, а также выделения волн различной длины из общего сигнала.

Для выделения волн в мультиплексоре могут использоваться разнообразные оптические механизмы. В оптических мультиплексорах, поддерживающих сравнительно небольшое количество длин волн в волокне, обычно 16 или 32, применяются **тонкопленочные фильтры**. Они состоят из пластин с многослойным покрытием, в качестве такой пластины на практике применяется торец оптического волокна, скошенный под углом 30–45°, с нанесенным на него слоями покрытия. Для систем с большим числом волн требуются другие принципы фильтрации и мультиплексирования.

В мультиплексорах DWDM применяются интегрально выполненные **дифракционные фазовые решетки**, или **дифракционные структуры** (Arrayed Waveguide Grating, AWG). Функции пластин в них выполняют оптические волноводы или волокна. Приходящий мультиплексный сигнал попадает на входной порт (рис. 11.19, *а*). Затем этот сигнал проходит через волновод-пластину и распределяется по множеству волноводов, представляющих дифракционную структуру AWG. Сигнал в каждом из волноводов по-прежнему является мультиплексным, а каждый канал (λ_1, λ_2, ..., λ_n) остается представленным во всех волноводах. Далее происходит отражение сигналов от зеркальной поверхности, и в итоге световые потоки вновь собираются в волноводе-пластине, где происходит их фокусировка и интерференция — образуются пространственно разнесенные интерференционные максимумы интенсивности, соответствующие разным каналам. Геометрия волновода-пластины, в частности расположение выходных полюсов, и значения длины волноводов структуры AWG рассчитываются таким образом, чтобы интерференционные максимумы совпадали с выходными полюсами. Мультиплексирование происходит обратным путем.

Другой способ построения мультиплексора базируется не на одной, а на паре волноводов-пластин (рис. 11.19, *б*). Принцип действия такого устройства аналогичен предыдущему случаю за исключением того, что здесь для фокусировки и интерференции используется дополнительная пластина.

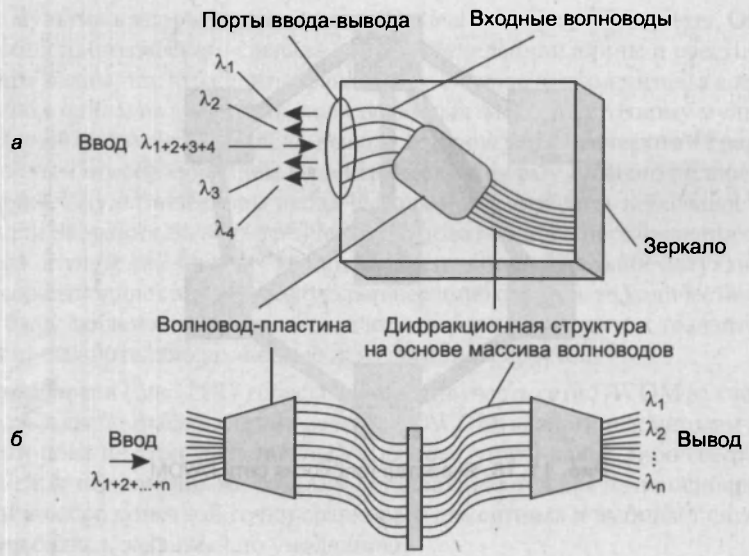


Рис. 11.19. Полное демультиплексирование сигнала с помощью дифракционной фазовой решетки

Интегральные решетки AWG (называемые также **фазарами**) стали одними из ключевых элементов мультиплексоров DWDM. Они обычно применяются для полного демультиплексирования светового сигнала, так как хорошо масштабируются и потенциально могут успешно работать в системах с сотнями спектральных каналов.

Оптические кросс-коннекторы

В сетях с ячеистой топологией необходимо обеспечить гибкие возможности для изменения маршрута следования волновых соединений между абонентами сети. Такие возможности предоставляют оптические кросс-коннекторы, позволяющие направить любую из волн входного сигнала каждого порта в любой из выходных портов (конечно, при условии, что никакой другой сигнал этого порта не использует эту волну, иначе необходимо выполнить трансляцию длины волны).

Существуют оптические кросс-коннекторы двух типов:

- **оптоэлектронные кросс-коннекторы** с промежуточным преобразованием в электрическую форму;
- **полностью оптические кросс-коннекторы**, или **фотонные коммутаторы**.

Исторически первыми появились оптоэлектронные кросс-коннекторы, за которыми и закрепилось название оптических кросс-коннекторов. Поэтому производители полностью оптических устройств этого типа стараются использовать для них другие названия: фотонные коммутаторы, маршрутизаторы волн, лямбда-маршрутизаторы. У оптоэлектронных кросс-коннекторов имеется принципиальное ограничение — они хорошо справляются со своими обязанностями при работе на скоростях до 2,5 Гбит/с, но начиная со скорости 10 Гбит/с и выше, габариты таких устройств и потребление энергии превышают допустимые пределы. Фотонные коммутаторы свободны от такого ограничения.

В фотонных коммутаторах используются различные оптические механизмы, в том числе дифракционные фазовые решетки и **микроэлектронные механические системы** (Micro-Electro Mechanical System, MEMS).

MEMS представляет собой набор подвижных зеркал очень маленького (диаметром менее миллиметра) размера (рис. 11.20). Коммутатор на основе MEMS включается в работу после демультиплексора, когда исходный сигнал уже разделен на составляющие волны. За счет поворота микрозеркала на заданный угол исходный луч определенной волны направляется в соответствующее выходное волокно. Затем все лучи мультиплексируются в общий выходной сигнал.

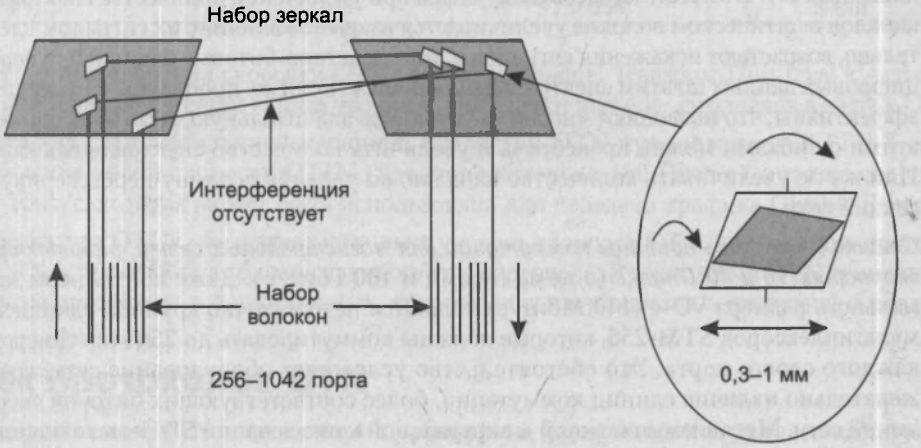


Рис. 11.20. Микроэлектронная механическая система кросс-коммутации

По сравнению с оптоэлектронными кросс-коннекторами фотонные коммутаторы занимают объем в 30 раз меньше и потребляют примерно в 100 раз меньше энергии. Однако у устройств этого типа низкое быстродействие, к тому же они чувствительны к вибрации. Тем не менее системы MEMS находят широкое применение в новых моделях фотонных коммутаторов. Сегодня подобные устройства могут обеспечивать коммутацию 256 × 256 спектральных каналов, планируется выпуск устройств с возможностями коммутации 1024 × 1024 каналов и выше.

Сети OTN

Причины и цели создания

Сети DWDM не являются собственно цифровыми сетями, так как они лишь предоставляют пользователям отдельные спектральные каналы, являющиеся не более чем несущей средой. Для того чтобы передавать по такому каналу цифровые данные, необходимо каким-то образом договориться о методе модуляции или кодирования двоичных данных, а также предусмотреть такие важные механизмы, как контроль корректности данных, исправление битовых ошибок, обеспечение отказоустойчивости, оповещение пользователя о состоянии соединения и т. п.

Исторически мультиплексоры DWDM были также и мультиплексорами SDH, то есть в каждом из волновых каналов для решения перечисленных задач они использовали технику SDH. Однако по прошествии некоторого времени эксплуатации сетей SDH/DWDM стали заметны определенные недостатки, связанные с применением технологии SDH в качестве основной технологии передачи цифровых данных по спектральным каналам DWDM.

Перечислим эти недостатки.

- *Недостаточная эффективность кодов FEC, принятых в качестве стандарта SDH.* Это препятствует дальнейшему повышению плотности спектральных каналов в мультиплексорах DWDM. Логика здесь следующая: при увеличении количества спектральных каналов в оптическом волокне увеличивается взаимное влияние их сигналов, следовательно, возрастают искажения сигналов и, как следствие, битовые ошибки при передаче цифровых данных по этим спектральным каналам. Если же процедуры FEC настолько эффективны, что позволяют «на лету» устранить значительную часть этих ошибок, то этими ошибками можно пренебречь и увеличить количество спектральных каналов. Или же не увеличивать количество каналов, но увеличить длину нерегенерируемых секций сети.
- *Слишком «мелкие» единицы коммутации для магистральных сетей, работающих на скоростях 10 и 40 Гбит/с* (а не за горами и 100 Гбит/с). Даже контейнеры максимального размера VC-4 (140 Мбит/с) являются недостаточно крупной единицей для мультиплексоров STM-256, которые должны коммутировать до 256 контейнеров для каждого своего порта. Это обстоятельство усложняет оборудование сети, поэтому желательно наличие единиц коммутации, более соответствующих битовой скорости линий сети. Механизмы смежной и виртуальной конкатенации SDH частично решают эту проблему, но она остается.
- *Не учтены особенности трафика различного типа.* Разработчиками технологии SDH принимался во внимание только голосовой трафик.

На преодоление этих недостатков нацелена новая технология **оптических транспортных сетей** (Optical Transport Network, OTN), которая обеспечивает передачу и мультиплексирование цифровых данных по волновым каналам DWDM более эффективно, чем SDH. В то же время сети OTN обеспечивают обратную совместимость с SDH, так как для мультиплексоров OTN трафик SDH является одним из видов пользовательского трафика наряду с такими клиентами, как Ethernet и GFP.

Нужно отметить, что технология OTN не заменяет технологии DWDM, а дополняет ее волновые каналы «цифровой оболочкой»[1].

Архитектура сетей OTN описана в стандарте ITU-T G.872, а наиболее важные технические аспекты работы узла сети OTN описаны в стандарте G.709.

Иерархия скоростей

Технология OTN многое взяла от технологии SDH, в том числе коэффициент кратности скоростей 4 для построения своей иерархии скоростей. Однако начальная скорость иерархии скоростей OTN гораздо выше, чем у SDH: 2,5 Гбит/с вместо 155 Мбит/с.

---

[1] Термин «цифровая оболочка» (digital wrapper) иногда даже используется в качестве названия самой технологии OTN.

В настоящее время стандартизованы три скорости OTN, которые выбраны так, чтобы прозрачным образом передавать кадры STM-16, STM-64 и STM-256 вместе со служебными заголовками (табл. 11.4).

Таблица 11.4. Иерархия скоростей технологии OTN

Интерфейс G.709	Битовая скорость, Гбит/с	Соответствующий уровень SDH	Битовая скорость, Гбит/с
OTU1	2,666	STM-16	2,488
OTU2	10,709	STM-64	9,953
OTU3	43,018	STM-256	39,813

Приведенные значения скорости OTUk (Optical Channel Transport Unit level k — транспортный блок оптического канала уровня *k*) учитывают наличие заголовков в кадрах OTN.

Работа над стандартизацией иерархии скоростей OTN продолжается, в ITU-T идет обсуждение новой скорости OTU4 (предположительно 160 Гбит/с), а также скорости в 1,2 Гбит/с, которая может быть использована для передачи трафика Gigabit Ethernet.

Аббревиатура OTUk обозначает не только уровень скорости OTN, но и один из протоколов OTN, а также формат блоков данных этого протокола. В технологии OTN существуют и другие протоколы и блоки данных, которые рассматриваются в следующем разделе.

Стек протоколов OTN

Стек протоколов OTN состоит из 4-х уровней, их назначение напоминает назначение уровней стека протоколов SDH.

На рис. 11.21 показана обобщенная архитектура сети OTN и области применения протокола каждого уровня, а на рис.11.22 — иерархия протоколов OTN.

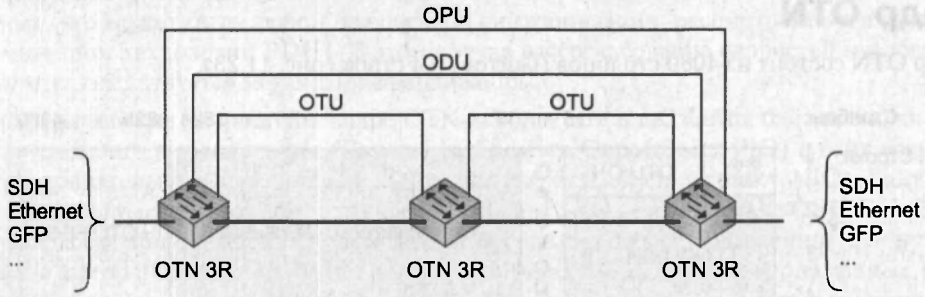


Рис. 11.21. Сеть OTN и распределение протоколов

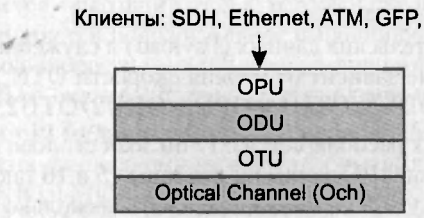


Рис. 11.22. Иерархия протоколов OTN

Нижний уровень протоколов составляет **оптический канал** (Optical Channel, Och); обычно это спектральный канал DWDM. Данный уровень примерно соответствует фотонному уровню технологии SDH.

Протокол OPU (Optical Channel Payload Unit — блок пользовательских данных оптического канала) ответственен за доставку данных между пользователями сети. Он занимается инкапсуляцией пользовательских данных, таких как кадры SDH или Ethernet, в блоки OPU, выравниванием скоростей передачи пользовательских данных и блоков OPU, а на приемной стороне извлекает пользовательские данные и передает их пользователю. В зависимости от скорости передачи данных этому протоколу соответствуют блоки OPU1, OPU2 и OPU3. Для выполнения своих функций протокол OPU добавляет к пользовательским данным свой заголовок OPU OH (OverHead). Блоки OPU не модифицируются сетью. Этот протокол является аналогом протокола тракта SDH.

Протокол ODU (Optical Channel Data Unit — блок данных оптического канала) так же, как и протокол OPU, работает между конечными узлами сети OTN. В его функции входит мультиплексирование и демультиплексирование блоков OPU, то есть, например, мультиплексирование четырех блоков OPU1 в один блок OPU2. Кроме того, протокол ODU поддерживает функции мониторинга качества соединений в сети OTN. Этот протокол формирует блоки ODU соответствующей скорости, добавляя к соответствующим блокам OPU свой заголовок. Протокол ODU является аналогом протокола линии SDH.

Протокол OTU (Optical Channel Transport Unit — транспортный блок оптического канала) работает между двумя соседними узлами сети OTN, которые поддерживают функции электрической регенерации оптического сигнала, называемые также функциями 3R (retiming, reshaping и regeneration). Основное назначение этого протокола — контроль и исправление ошибок с помощью кодов FEC. Этот протокол добавляет к блоку ODUk свой концевик, содержащий код FEC, образуя блок OTUk. Протокол OTU соответствует протоколу секции SDH. Блоки OTUk помещаются непосредственно в оптический канал.

Кадр OTN

Кадр OTN состоит из 4080 столбцов (байтов) и 4 строк (рис. 11.23).

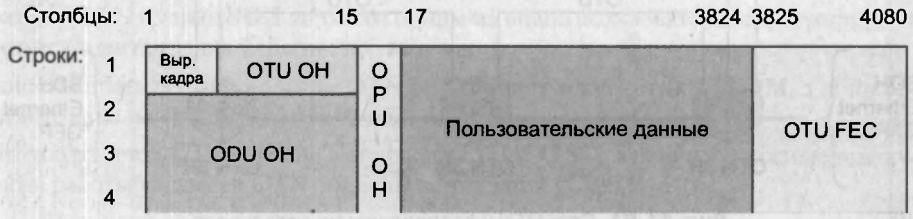


Рис. 11.23. Формат кадра OTN

Кадр состоит из поля пользовательских данных (Payload) и служебных полей блоков OPU, ODU и OTU. Формат кадра не зависит от уровня скорости OTN, то есть он, например, одинаков для блоков OPU1/ODU1/OTU1 и OPU2/ODU2/OTU2.

Поле пользовательских данных располагается с 17 по 3824 столбец и занимает все четыре строки кадра, а заголовок блока OPU занимает столбцы 15 и 16 также в четырех строках. При необходимости заголовок OPU OH может занимать несколько кадров подряд (в этих

случаях говорят о мультикадре OTN), например, такой вариант встречается в том случае, когда нужно описать структуру поля пользовательских данных, мультиплексирующую несколько блоков OPU более низкого уровня.

Блок ODU представлен только заголовком ODU OH (формально он также имеет поле данных, в которое помещен блок OPU), а блок OTU состоит из заголовка OTU OH и концевика OTU FEC, содержащего код коррекции ошибок FEC.

Начинается кадр с небольшого поля выравнивания кадра, необходимого для распознавания начала кадра.

Выравнивание скоростей

Как и в других технологиях, основанных на синхронном мультиплексировании TDM, в технологии OTN решается проблема выравнивания скоростей пользовательских потоков со скоростью передачи данных мультиплексора. Механизм выравнивания скоростей OTN является некоторым гибридом механизма бит-стаффинга технологии PDH и механизма положительного и отрицательного выравнивания на основе указателей, используемого в технологии SDH.

Работа механизма выравнивания OTN зависит от того, какой режим отображения нагрузки на кадры OTM поддерживается для данного пользовательского потока — синхронный или асинхронный. В режиме **синхронного отображения нагрузки** мультиплексор OTM синхронизирует прием и передачу данных от синхроимпульсов, находящихся в принимаемом потоке пользовательских данных. Этот режим рассчитан на пользовательские протоколы, данные которых хорошо синхронизированы и содержат в заголовке специальные биты синхронизации (такие как SDH). В этом случае механизм выравнивания фактически простаивает, так как скорость передачи данных всегда равна скорости их поступления, поэтому выравнивать нечего.

В режиме **асинхронного отображения нагрузки** мультиплексор OTN синхронизируется от собственного источника синхроимпульсов, который не зависит от пользовательских данных (это может быть любой из способов синхронизации, рассмотренных в разделе, посвященном технологии PDH). В этом случае рассогласование скоростей неизбежно, и поэтому задействуется механизм выравнивания.

Для выравнивания скоростей в кадре OTN используются два байта: байт возможности положительного выравнивания (Positive Justification Opportunity, PJO) и байт возможности отрицательного выравнивания (Negative Justification Opportunity, NJO). Байт PJO находится в поле пользовательских данных, а байт NJO — в заголовке OPU OH. В тех случаях, когда при помещении пользовательских данных скорость выравнивать не нужно, мультиплексор помещает все байты пользовательских данных в байты поля данных, применяя в том числе и байт PJO. В тех случаях, когда скорость пользовательского потока меньше скорости мультиплексора и ему не хватает байта для заполнения поля данных, то в байт PJO вставляется «выравниватель», который представляет собой байт с нулевым значением — так выполняется положительное выравнивание. А если скорость пользовательского потока больше скорости мультиплексора, лишний байт пользовательских данных помещается в поле NJO — так происходит отрицательное выравнивание.

Для того чтобы конечный мультиплексор сети правильно выполнил демультиплексирование пользовательских данных, ему нужна информация о том, каким образом в кадре использованы байты NJO и PJO. Такая информация хранится в поле управления выравни-

ванием (Justification Control, JC), два бита которого показывают, какое значение помещено в каждый из байтов NJO и PJO.

Указатель на начало пользовательских данных в технологии OTN не задействован. Таким образом, вставка байта делает механизм выравнивания OTN похожим на PDH, где имеет место вставка битов и соответствующие признаки такой вставки (отрицательное выравнивание). С технологией SDH механизм выравнивания OTN роднит применение как отрицательного, так и положительного выравнивания байтами.

Мультиплексирование блоков

При мультиплексировании блоков ODU поле пользовательских данных блока OPUk разбивается на так называемые **трибутарные слоты** (Tributary Slot, TS), в которые помещаются данные блока OPUk-1.

На рис. 11.24 показан пример мультиплексирования четырех блоков ODU1 в один блок ODU2. Как видно из рисунка, поле данных блока OPU2 разбито на трибутарные слоты TribSlot1, TribSlot2, TribSlot3 и TribSlot4, последовательность которых повторяется. Каждый из этих четырех трибутарных слотов предназначен для переноса части поля данных одного из блоков OPU1. Здесь используется техника чередования данных скорости более низкого уровня иерархии скоростей в поле данных блока более высокой скорости иерархии скоростей, которая типична для технологий синхронного временного мультиплексирования. Эта техника обеспечивает выполнение операций мультиплексирования и демультиплексирования «на лету» без промежуточной буферизации, так как частота появления порций данных OPU1 в блоке ODU2 соответствует частоте их появления в том случае, если бы они передавались на скорости OPU1.

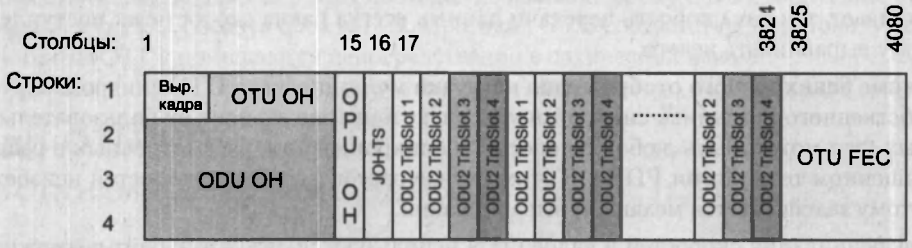


Рис. 11.24. Мультиплексирование блоков ODU1в блок ODU2

Техника мультиплексирования блоков ODU1 и ODU2 в блок ODU3 аналогична, если не считать того, что в блоке OPU3 используется 16 различных трибутарных слотов, что позволяет поместить в него 16 блоков ODU1 или 4 блока ODU2 (в этом случае одной порции OPU2 соответствует четыре трибутарных слота ODU3).

Информация об использовании трибутарных слотов хранится в специальном разделе поля OPU2 OH или OPU3 OH. Этот раздел может также запоминать информацию о *виртуальной конкатенации блоков ODU1 или ODU2* — эта техника также поддерживается в сетях OTN.

Коррекция ошибок

В OTN применяется процедура прямой коррекции ошибок (FEC), в которой используются коды Рида—Соломона RS(255, 239). В этом самокорректирующемся коде данные кодиру-

ются блоками по 255 байт, из которых 239 байт являются пользовательскими, а 16 байт представляют собой корректирующий код. Коды Рида—Соломона позволяют исправлять до 8 ошибочных байт в блоке из 255 байт, что является очень хорошей характеристикой для самокорректирующего кода.

Применение кода Рида—Соломона позволяет улучшить отношение мощности сигнала к мощности шума на 5 дБ при уровне битовых ошибок в 10^{-12}. Этот эффект дает возможность увеличить расстояние между регенераторами сети на 20 км или использовать менее мощные передатчики сигнала.

Выводы

Первичные сети предназначены для создания коммутируемой инфраструктуры, с помощью которой можно достаточно быстро создать постоянные каналы, организующие произвольную топологию.

В первичных сетях используют технику коммутации каналов различного типа: с частотным (FDM), временны́м (TDM) и волновым (WDM/DWDM) мультиплексированием.

В сетях FDM каждому абонентскому каналу выделяется полоса частот шириной 4 кГц. Существует иерархия каналов FDM, при этом 12 абонентских каналов образуют группу каналов первого уровня иерархии (базовую группу) с полосой 48 кГц, 5 каналов первого уровня объединяются в канал второго уровня иерархии (супергруппу) с полосой 240 кГц, а 10 каналов второго уровня составляют канал третьего уровня иерархии (главную группу) с полосой в 2,4 МГц.

Цифровые первичные сети PDH позволяют образовывать каналы с пропускной способностью от 64 Кбит/с до 140 Мбит/с, предоставляя своим абонентам скорости четырех уровней иерархии.

Недостатком сетей PDH является невозможность непосредственного выделения данных низкоскоростного канала из данных высокоскоростного канала, если каналы работают на несмежных уровнях иерархии скоростей.

Асинхронность ввода абонентских потоков в кадр SDH обеспечивается благодаря концепции виртуальных контейнеров и системы плавающих указателей, отмечающих начало пользовательских данных в виртуальном контейнере.

Мультиплексоры SDH могут работать в сетях с различной топологией (цепи, кольца, ячеистая топология). Различают несколько специальных типов мультиплексоров, которые занимают особое место в сети: терминальные мультиплексоры, мультиплексоры ввода-вывода, кросс-коннекторы.

В сетях SDH поддерживается большое количество механизмов отказоустойчивости, которые защищают трафик данных на уровне отдельных блоков, портов или соединений: EPS, CP, MSP, SNC-P и MS-SPRing. Наиболее эффективная схема защиты выбирается в зависимости от логической топологии соединений в сети.

Технология WDM/DWDM реализует принципы частотного мультиплексирования для сигналов иной физической природы и на новом уровне иерархии скоростей. Каждый канал WDM/DWDM представляет собой определенный диапазон световых волн, позволяющих переносить данные в аналоговой и цифровой форме, при этом полоса пропускания канала в 25–50–100 ГГц обеспечивает скорости в несколько гигабит в секунду (при передаче дискретных данных).

В ранних системах WDM использовалось небольшое количество спектральных каналов, от 2 до 16. В системах DWDM задействовано уже от 32 до 160 каналов на одном оптическом волокне, что обеспечивает скорости передачи данных для одного волокна до нескольких терабит в секунду.

Современные оптические усилители позволяют удлинить оптический участок линии связи (без преобразования сигнала в электрическую форму) до 700–1000 км.

Для выделения нескольких каналов из общего светового сигнала разработаны сравнительно недорогие устройства, которые обычно объединяются с оптическими усилителями для организации мультиплексоров ввода-вывода в сетях дальней связи.

Для взаимодействия с традиционными оптическими сетями (SDH, Gigabit Ethernet, 10G Ethernet) в сетях DWDM применяются транспондеры и трансляторы длин волн, которые преобразуют длину волны входного сигнала в длину одной из волн стандартного частотного плана DWDM.

В полностью оптических сетях все операции мультиплексирования и коммутации каналов выполняются над световыми сигналами без их промежуточного преобразования в электрическую форму. Это упрощает и удешевляет сеть.

Технология OTN позволяет более эффективно использовать спектральные каналы сетей DWDM, поддерживая экономные схемы мультиплексирования данных на высоких скоростях.

Мощный механизм коррекции ошибок OTN FEC, использующий самокорректирующиеся коды Рида—Соломона, позволяет улучшить отношение сигнал/шум в спектральных каналах и увеличить расстояние между регенераторами сети.

Вопросы и задания

1. Какие недостатки первичных сетей FDM привели к созданию цифровых первичных сетей?
2. Название T-1 обозначает:
   а) аппаратуру мультиплексирования, разработанную компанией AT&T;
   б) уровень скорости 1,544 Мбит/с;
   в) международный стандарт линии связи;
   г) способ мультиплексирования цифровых потоков 64 Кбит/с.
3. Какие функции выполняет младший бит каждого байта в канале T-1 при передаче голоса?
4. Можно ли в сети PDH выделить канал DS-0 непосредственно из канала DS-3?
5. Какие механизмы в канале E-1 заменяют «кражу бита» канала T-1?
6. Почему первичные сети обеспечивают высокое качество обслуживания всех видов трафика?
7. Какое свойство технологии PDH отражает слово «плезиохронная»?
8. Каким образом компенсируется отсутствие синхронности трибутарных потоков в технологии SDH?
9. Какое максимальное количество каналов E-1 может мультиплексировать кадр STM-1?
10. Сколько каналов T-1 может мультиплексировать кадр STM-1, если в нем уже мультиплексировано 15 каналов E-1?
11. По какой причине в кадре STM-1 используется три указателя?
12. С какой целью в технологиях PDH и SDH применяется чередование байтов?
13. В чем отличие схем защиты 1+1 и 1:1? Варианты ответов:
    а) в схеме 1+1 два потока мультиплексируются в один, а в схеме 1:1 нет;
    б) схема 1+1 говорит о том, что резервный элемент выполняет те же функции, что и основной, а в схеме 1:1 резервный элемент простаивает до момента выхода из строя основного;
    в) схема 1+1 используется для защиты портов, а схема 1:1 — для защиты путей трафика.
14. При каких условиях защита MS-SPRing более эффективна, чем SNC-P?

15. Для достижения каких целей разработан механизм виртуальной конкатенации? Варианты ответов:

    а) для эффективной передачи трафика телефонных сетей;

    б) для эффективной передачи трафика Ethernet;

    в) для повышения верхней границы скоростей технологии SDH.

16. Можно ли объединять контейнеры VC-3 за счет смежной конкатенации?
17. Можно ли передавать составляющие контейнеры при виртуальной конкатенации по разным маршрутам?
18. Можно ли динамически изменить пропускную способность соединения SDH?
19. Почему протокол GFP в режиме GFP-F не использует пустые кадры для выравнивания скоростей?
20. Что общего между первичными сетями FDM и DWDM?
21. К какому типу сетей относятся сети DWDM, аналоговым или цифровым?
22. С какой целью в сетях DWDM используются регенераторы, преобразующие оптический сигнал в электрический?
23. Назовите причины ухудшения качества оптического сигнала при передаче через большое количество пассивных участков DWDM?
24. С какой частотой будет выполняться операция отрицательного выравнивания указателя контейнера VC-4 в кадре STM-1, если относительная разница между тактовыми частотами передающего и принимающего мультиплексоров SDH равна 10^{-5}?
25. Какие недостатки технологии SDH послужили причиной создания новой технологии OTN? Варианты ответов:

    а) недостаточная гибкость механизма указателей;

    б) слишком мелкие единицы коммутации;

    в) низкая эффективность кодов FEC.

Часть III

Локальные вычислительные сети

Локальные сети являются неотъемлемой частью любой современной компьютерной сети. Если мы рассмотрим структуру глобальной сети, например Интернета или крупной корпоративной сети, то обнаружим, что практически все информационные ресурсы этой сети сосредоточены в локальных сетях, а глобальная сеть является транспортом, который соединяет многочисленные локальные сети.

Технологии локальных сетей прошли большой путь. Практически во всех технологиях 80-х годов использовалась *разделяемая среда* как удобное и экономичное средство объединения компьютеров на физическом уровне. С середины 90-х в локальных сетях стали также применяться *коммутируемые* версии технологий. Отказ от разделяемой среды позволил повысить производительность и масштабируемость локальных сетей. Преимуществом коммутируемых локальных сетей является также возможность логической структуризации сети с разделением ее на отдельные сегменты, называемые виртуальными локальными сетями.

Переход к коммутируемым локальным средам сопровождался победой одной технологии, а именно технологии Ethernet. Остальные технологии, такие как Arcnet, Token Ring и FDDI, остались в прошлом, несмотря даже на то, что они обладали хорошими техническими характеристиками и имели многочисленных пользователей.

Неизвестно, что больше повлияло на такую ситуацию, то ли предельная простота технологии, а значит, и низкая стоимость оборудования Ethernet и его эксплуатации, то ли удачное название, то ли просто необыкновенное везение, как считает изобретатель этой технологии Роберт Меткалф, состоявшее в том, что «каждый раз, когда появлялось что-то на замену Ethernet, люди, ответственные за продвижение новой технологии, снова выбирали для нее название Ethernet», но факт остается фактом — локальные сети стали однородными сетями Ethernet.

В локальных сетях изменился не только принцип использования среды. Быстро растет верхний предел информационной скорости протоколов локальных сетей. С принятием в 2002 году стандарта 10G Ethernet технологии локальных сетей стали поддерживать иерархию скоростей, не уступающую иерархии скоростей первичных сетей — от 10 Мбит/с до 10 Гбит/с. Это дает возможность строить на этих технологиях не только локальные сети, но и сети мегаполисов. И не за горами принятие нового стандарта — 100G Ethernet, поддерживающего скорость 100 Гбит/с.

Развитие локальных сетей идет и в направлении «миниатюризации» — появился новый тип сетей — *персональные сети* (Personal Area Network, PAN), которые объединяют электронные устройства одного пользователя в радиусе нескольких десятков метров.

В главе 12 рассматриваются технологии локальных сетей на разделяемой среде: основное внимание уделено классическим вариантам Ethernet со скоростью 10 Мбит/с на коаксиале и витой паре; также здесь кратко рассмотрены принципы работы основных соперников Ethernet в 80-е и 90-е годы — технологий Token Ring и FDDI. Если проводные технологии локальных сетей на разделяемой среде интересны сегодня в основном в теоретическом плане (для понимания истоков и динамики развития современных технологий), то беспроводные технологии локальных сетей на основе разделяемой среды по-прежнему актуальны и, по всей видимости, останутся таковыми в обозримом будущем, так как радиоэфир является разделяемой средой по своей природе. Мы рассмотрим две наиболее популярные технологии этого семейства — IEEE 802.11 (LAN) и Bluetooth (PAN).

Глава 13 посвящена коммутируемым локальным сетям. В ней рассматриваются основные принципы работы таких сетей: алгоритм функционирования коммутатора локальной сети, дуплексные версии протоколов локальных сетей, особенности реализации коммутаторов локальных сетей.

В главе 14 изучаются расширенные возможности коммутируемых локальных сетей этого типа: резервные связи на основе алгоритма покрывающего дерева, агрегирование каналов, а также техника виртуальных локальных сетей, позволяющая быстро и эффективно выполнять логическую структуризацию сети.

- Глава 12. Технологии локальных сетей на разделяемой среде
- Глава 13. Коммутируемые сети Ethernet
- Глава 14. Интеллектуальные функции коммутаторов

ГЛАВА 12 Технологии локальных сетей на разделяемой среде

Алгоритм доступа к разделяемой среде является одним из главных факторов, определяющих эффективность совместного использования среды конечными узлами локальной сети. Можно сказать, что алгоритм доступа формирует «облик» технологии, позволяет отличать данную технологию от других. В технологии Ethernet применяется очень простой алгоритм доступа, позволяющий узлу сети передавать данные в те моменты времени, когда он считает, что разделяемая среда свободна. Простота алгоритма доступа определила простоту и низкую стоимость оборудования Ethernet. Негативным атрибутом алгоритма доступа технологии Ethernet являются коллизии, то есть ситуации, когда кадры, передаваемые разными станциями, сталкиваются друг с другом в общей среде. Коллизии снижают эффективность разделяемой среды и придают работе сети непредсказуемый характер.

Первоначальный вариант технологии Ethernet был рассчитан на коаксиальный кабель, который использовался всеми узлами сети в качестве общей шины. Переход на кабельные системы на витой паре и концентраторах (хабах) существенно повысил эксплуатационные характеристики сетей Ethernet.

В технологиях Token Ring и FDDI поддерживались более сложные и эффективные алгоритмы доступа к среде, основанные на передаче друг другу токена — специального кадра, разрешающего доступ. Однако чтобы выжить в конкурентной борьбе с Ethernet, этого преимущества оказалось недостаточно.

Общая характеристика протоколов локальных сетей на разделяемой среде

Стандартная топология и разделяемая среда

Основная цель, которую ставили перед собой разработчики первых локальных сетей во второй половине 70-х годов, заключалась в нахождении простого и дешевого решения для объединения в вычислительную сеть нескольких десятков компьютеров, находящихся в пределах одного здания. Решение должно было быть недорогим, поскольку компьютеры, объединявшиеся в сеть, были недороги — появившиеся и быстро распространявшиеся тогда мини-компьютеры стоимостью в 10 000–20 000 долларов. Количество их в одной организации было небольшим, поэтому предел в несколько десятков компьютеров представлялся вполне достаточным для практически любой локальной сети. Задача связи локальных сетей в глобальные не была первоочередной, поэтому практически все технологии локальных сетей ее игнорировали.

Для упрощения и, соответственно, удешевления аппаратных и программных решений разработчики первых локальных сетей остановились на совместном использовании **общей среды передачи данных**.

Этот метод связи компьютеров впервые был опробован при создании радиосети ALOHA Гавайского университета в начале 70-х под руководством Нормана Абрамсона (Norman Abramson). Радиоканал определенного диапазона частот естественным образом является общей средой для всех передатчиков, использующих частоты этого диапазона для кодирования данных. Сеть ALOHA работала по методу случайного доступа, когда каждый узел мог начать передачу пакета в любой момент времени. Если после этого он не дожидался подтверждения приема в течение определенного тайм-аута, он посылал этот пакет снова. Общим был радиоканал с несущей частотой 400 МГц и полосой 40 кГц, что обеспечивало передачу данных со скоростью 9600 бит/с.

Немного позже Роберт Меткалф (Robert Metcalfe) повторил идею разделяемой среды уже для проводного варианта технологии LAN. Непрерывный сегмент коаксиального кабеля стал аналогом общей радиосреды. Все компьютеры присоединялись к этому сегменту кабеля по схеме монтажного ИЛИ, поэтому при передаче сигналов одним из передатчиков все приемники получали один и тот же сигнал, как и при использовании радиоволн. На рис. 12.1 представлено начало служебной записки Роберта Меткалфа, написанной 22 мая 1973 года, с наброском разделяемой среды на коаксиальном кабеле, где эта среда названа «a cable-tree ether», что можно приблизительно перевести как «древовидный кабельный эфир».

В технологиях Token Ring и FDDI тот факт, что компьютеры используют разделяемую среду, не так очевиден, как в случае Ethernet. Физическая топология этих сетей — кольцо, каждый узел соединяется кабелем с двумя соседними узлами (рис. 12.2). Однако эти отрезки кабеля также являются разделяемыми, так как в каждый момент времени только один компьютер может задействовать кольцо для передачи своих пакетов.

Простые **стандартные топологии физических связей** (звезда у коаксиального кабеля Ethernet и кольцо у Token Ring и FDDI) обеспечивают простоту разделения кабельной среды.

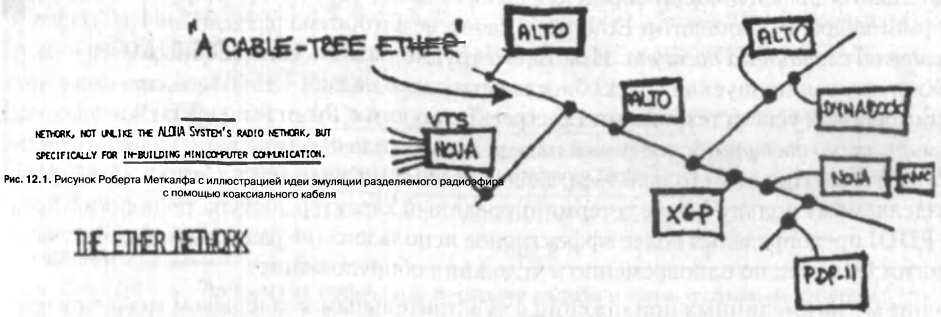


Рис. 12.1. Рисунок Роберта Меткалфа с иллюстрацией идеи эмуляции разделяемого радиоэфира с помощью коаксиального кабеля

Использование разделяемых сред позволяет *упростить* логику работы узлов сети. Действительно, поскольку в каждый момент времени выполняется только одна передача, отпадает необходимость в буферизации кадров в транзитных узлах и, как, следствие, в самих транзитных узлах. Соответственно, отпадает необходимость в сложных процедурах управления потоком и борьбы с перегрузками.

Основной недостаток разделяемой среды — плохая масштабируемость. Этот недостаток является принципиальным, так как независимо от метода доступа к среде ее пропускная способность делится между всеми узлами сети. Здесь применимо положение теории очередей, которое мы изучали в главе 7: как только коэффициент использования общей среды превышает определенный порог, очереди к среде начинают расти нелинейно, и сеть становится практически неработоспособной. Значение порога зависит от метода доступа. Так, в сетях ALOHA это значение является крайне низким — всего около 18 %, в сетях Ethernet — около 30 %, а в сетях Token Ring и FDDI оно возросло до 60–70 %.

Локальные сети, являясь пакетными сетями, используют принцип временного мультиплексирования, то есть разделяют передающую среду во времени. Алгоритм *управления доступом к среде* является одной из важнейших характеристик любой технологии LAN,

Рис. 12.2. Разделяемая среда в кольцевых топологиях

в значительно большей степени определяющей ее облик, чем метод кодирования сигналов или формат кадра. В технологии Ethernet в качестве алгоритма разделения среды применяется *метод случайного доступа*. И хотя его трудно назвать совершенным — при росте нагрузки полезная пропускная способность сети резко падает — он благодаря своей простоте стал основой успеха технологии Ethernet. Технологии Token Ring и FDDI используют *метод маркерного доступа*, основанный на передаче от узла к узлу особого кадра — маркера (токена) доступа. При этом только узел, владеющий маркером доступа, имеет право доступа к разделяемому кольцу. Более детерминированный характер доступа технологий Token Ring и FDDI предопределил более эффективное использование разделяемой среды, чем у технологии Ethernet, но одновременно и усложнил оборудование.

Появление мультимедийных приложений с чувствительным к задержкам трафиком привело к попыткам создания метода доступа, приоритезирующего некоторым образом такой трафик и обеспечивающего для него необходимые характеристики QoS. Результатом этих попыток стало создание технологии 100VG-AnyLAN, для которой был характерен достаточно сложный метод доступа к разделяемой среде. Однако эта технология была создана слишком поздно — в середине 90-х годов, когда преимущества и доступность коммутируемых локальных сетей «отменили» сам принцип разделения среды (в проводных сетях).

Отказ от разделяемой среды привел к исчезновению такого важного компонента технологии локальных сетей как метод доступа. В принципе коммутатор локальной сети работает так же, как и обобщенный коммутатор сети с коммутацией пакетов, рассмотренный в главе 2. Поэтому с распространением коммутаторов стали исчезать различия между технологиями локальных сетей, так как в сети, где все связи между узлами являются индивидуальными, и коммутируемая версия Ethernet, и коммутируемая версия Token Ring работают весьма схоже, различаются только форматы кадров этих технологий. Это обстоятельство, возможно, и имел в виду Роберт Меткалф, когда говорил об удачливости Ethernet — работа коммутируемых локальных сетей Etherhet существенно отличается от работы Etherhet на разделяемой среде, так что ее можно считать новой технологией со старым названием. Хотя, с другой стороны, формат кадра Ethernet сохранился, так что это дает формальный (хотя и несколько условный) повод считать ее той же самой технологией.

Стандартизация протоколов локальных сетей

Каждая из технологий локальных сетей первоначально появлялась как фирменная технология; так, например, технология Ethernet «появилась на свет» в компании Xerox, а за технологией Token Ring стояла компания IBM. Первые стандарты технологий локальных сетей также были фирменными, что было, естественно, не очень удобно как для пользователей, так и для компаний-производителей сетевого оборудования.

Для исправления ситуации в 1980 году в институте IEEE был организован комитет 802 по стандартизации технологий LAN. Результатом работы комитета IEEE 802 стало принятие семейства стандартов IEEE 802.x, содержащих рекомендации по проектированию нижних уровней локальных сетей. Эти стандарты базировались на обобщении популярных фирменных стандартов, в частности Ethernet и Token Ring.

Комитет IEEE 802 и сегодня является основным международным органом, разрабатывающим стандарты технологий локальных сетей, в том числе коммутируемых локальных сетей, а также стандарты беспроводных локальных сетей на разделяемой среде.

Помимо IEEE в работе по стандартизации протоколов LAN принимали и принимают участие и другие организации. Так, для сетей, работающих на оптоволокне, институтом ANSI

был разработан стандарт FDDI, обеспечивающий скорость передачи данных 100 Мбит/с. Это был первый протокол LAN, который достиг такой скорости, в 10 раз превысив скорость технологии Ethernet.

Структуру стандартов IEEE 802 иллюстрирует рис. 12.3.

Рис. 12.3. Структура стандартов IEEE 802.x

Стандарты IEEE 802 описывают функции, которые можно отнести к функциям физического и канального уровней модели OSI. Как видно из рисунка 12.3, эти стандарты имеют как общие для всех технологий части, так и индивидуальные.

Общую группу стандартов составляют *стандарты рабочей группы 802.1*. Эти стандарты описывают наиболее высокоуровневые функции локальных сетей. Так, в документах 802.1 даются общие определения локальных сетей и их свойств, показана связь трех уровней модели IEEE 802 с моделью OSI. Наиболее практически важными являются те стандарты рабочей группы 802.1, которые описывают взаимодействие различных технологий, а также стандарты по построению более сложных сетей на основе базовых топологий. Эта группа стандартов носит общее название **стандартов межсетевого взаимодействия**. Наиболее важным в настоящее время является стандарт 802.1D, описывающий логику работы прозрачного моста, которая лежит в основе любого современного коммутатора Ethernet (и лежала бы в основе коммутатора Token Ring или FDDI, если бы они сохранились до наших дней).

Набор стандартов, разработанных рабочей группой 802.1, продолжает расти, в настоящее время это наиболее активный подкомитет комитета 802. Например, этот комитет стандартизовал технологию виртуальных локальных сетей, также он занимается стандартизацией технологий, известных под общим названием Carrier Ethernet.

Каждая из рабочих групп 802.3, 802.4, 802.5 и т. д. ответственна за стандартизацию конкретной технологии, например группа 802.3 занимается технологией Ethernet, группа 802.4 – технологией ArcNet, группа 802.5 — технологией Token Ring, группа 802.11 — тех-

нологией беспроводных локальных сетей. Стандарты этих рабочих групп описывают как физический уровень (или несколько возможных физических уровней), так и канальный уровень конкретной технологии (последний включает описание метода доступа, используемого технологией). Основу стандарта 802.3 составила технология экспериментальной сети Ethernet Network, которую фирма Xerox разработала и реализовала в 1975 году. В 1980 году фирмы DEC, Intel и Xerox (сокращенно — DIX) совместно разработали и опубликовали стандарт Ethernet версии II для сети, построенной на основе коаксиального кабеля. Эту последнюю версию фирменного стандарта Ethernet называют стандартом Ethernet DIX, или Ethernet II. На базе стандарта Ethernet DIX был разработан стандарт IEEE 802.3, который во многом совпадает со своим предшественником.

Однако, как видно из рис. 12.3, помимо индивидуальных для каждой технологии уровней существует и общий уровень, который был стандартизован рабочей группой 802.2.

Появление этого уровня связано с тем, что комитет 802 разделил функции канального уровня модели OSI на два уровня:

- управление логическим каналом (Logical Link Control, LLC);
- управление доступом к среде (Media Access Control, MAC).

Основными функциями уровня MAC являются:

- обеспечение доступа к разделяемой среде;
- передача кадров между конечными узлами посредством функций и устройств физического уровня.

Если уровень MAC специфичен для каждой технологии и отражает различия в методах доступа к разделяемой среде, то уровень LLC представляет собой обобщение функций разных технологий по обеспечению передачи кадра с различными требованиями к надежности.

Логика образования общего для всех технологий уровня LLC заключается в следующем: после того как узел сети получил доступ к среде в соответствии с алгоритмом, специфическим для конкретной технологии, дальнейшие действия узла или узлов по обеспечению надежной передачи кадров от технологии не зависят.

Так как в зависимости от требований приложения может понадобиться разная степень надежности, то рабочая группа 802.2 определила три типа услуг:

- **Услуга LLC1** — это услуга *без установления соединения и без подтверждения получения данных*. LLC1 дает пользователю средства для передачи данных с минимумом издержек. В этом случае LLC поддерживает дейтаграммный режим работы, как и MAC, так что и технология LAN в целом работает в дейтаграммном режиме. Обычно эта процедура используется, когда такие функции, как восстановление данных после ошибок и упорядочивание данных, выполняются протоколами вышележащих уровней, поэтому нет нужды дублировать их на уровне LLC.
- **Услуга LLC2** дает пользователю возможность установить *логическое соединение* перед началом передачи любого блока данных и, если это требуется, выполнить *процедуры восстановления* после ошибок и упорядочивание потока блоков в рамках установленного соединения.
- **Услуга LLC3** — это услуга *без установления соединения, но с подтверждением получения данных*. В некоторых случаях (например, при использовании сетей в системах реального времени, управляющих промышленными объектами), с одной стороны, временные издержки установления логического соединения перед отправкой данных неприемлемы, а с другой стороны, подтверждение о корректности приема переданных данных необходимо. Для такого рода ситуаций и предусмотрена дополнительная услуга LLC3, которая

является компромиссом между LLC1 и LLC2, так как она не предусматривает установление логического соединения, но обеспечивает подтверждение получения данных.

Какой из трех режимов работы уровня LLC будет использован, зависит от требований протокола верхнего уровня. Информация о требуемой от LLC транспортной услуге передается через межуровневый интерфейс уровню LLC вместе с аппаратным адресом и пакетом с пользовательскими данными. Например, когда поверх LLC работает протокол IP, он всегда запрашивает режим LLC1, поскольку в стеке TCP/IP задачу обеспечения надежной доставки решает протокол TCP.

Нужно сказать, что на практике идея обобщения функций обеспечения надежной передачи кадров в общем уровне LLC не оправдала себя. Технология Ethernet в версии DIX изначально функционировала в наиболее простом дейтаграммном режиме — в результате оборудование Ethernet и после опубликования стандарта IEEE 802.2 продолжало поддерживать только этот режим работы, который формально является режимом LLC1. В то же время оборудование сетей Token Ring, которое изначально поддерживало режимы LLC2 и LLC3, также продолжало поддерживать эти режимы и никогда не поддерживало режим LLC1.

Помимо обеспечения заданной степени надежности уровень LLC выполняет также *интерфейсные функции*. Эти функции заключаются в передаче пользовательских и служебных данных между уровнем MAC и сетевым уровнем. При передаче данных *сверху вниз* уровень LLC принимает от протокола сетевого уровня пакет (например, IP- или IPX-пакет), в котором уже находятся пользовательские данные. Помимо пакета сверху также передается адрес узла назначения в формате той технологии LAN, которая будет использована для доставки кадра в пределах данной локальной сети. Напомним, что в терминах стека TCP/IP такой адрес называется аппаратным. Полученные от сетевого уровня пакет и аппаратный адрес уровень LLC передает далее вниз — уровню MAC. Кроме того, LLC при необходимости решает *задачу мультиплексирования*, передавая данные от нескольких протоколов сетевого уровня единственному протоколу уровня MAC.

При передаче данных *снизу вверх* LLC принимает от уровня MAC пакет сетевого уровня, пришедший из сети. Теперь ему нужно выполнить еще одну интерфейсную функцию — *демультиплексирование*, то есть решить, какому из сетевых протоколов передать полученные от MAC данные (рис. 12.4).

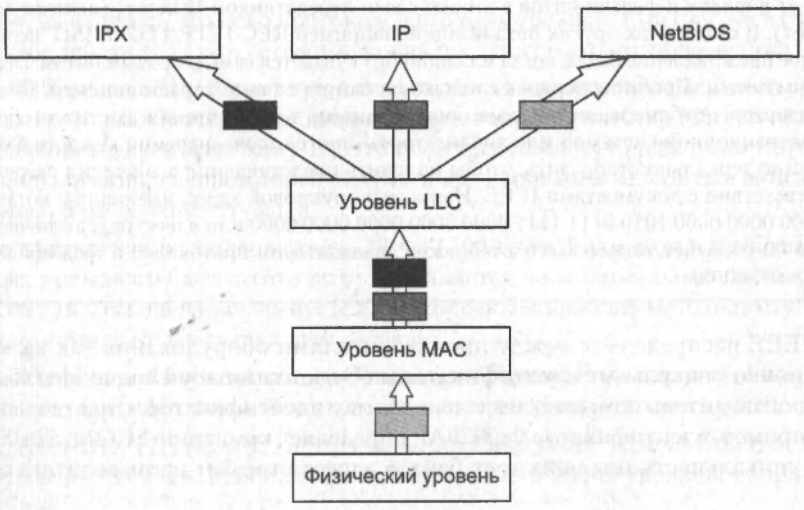


Рис. 12.4. Демультиплексирование кадров протоколом LLC

Ethernet со скоростью 10 Мбит/с на разделяемой среде

MAC-адреса

На уровне MAC, который обеспечивает доступ к среде и передачу кадра, для идентификации сетевых интерфейсов узлов сети используются регламентированные стандартом IEEE 802.3 уникальные 6-байтовые адреса, называемые **MAC-адресами**. Обычно MAC-адрес записывают в виде шести пар шестнадцатеричных цифр, разделенных тире или двоеточиями, например 11-A0-17-3D-BC-01. Каждый сетевой адаптер имеет, по крайней мере, один MAC-адрес.

Помимо отдельных интерфейсов, MAC-адрес может определять группу интерфейсов или даже все интерфейсы сети. Первый (младший) бит старшего байта адреса назначения является признаком того, является адрес индивидуальным или групповым. Если он равен 0, то адрес является **индивидуальным**, то есть идентифицирует один сетевой интерфейс, а если 1, то **групповым**. Групповой адрес связан только с интерфейсами, сконфигурированными (вручную или автоматически по запросу вышележащего уровня) как члены группы, номер которой указан в групповом адресе. Если сетевой интерфейс включен в группу, то наряду с уникальным MAC-адресом с ним ассоциируется еще один адрес — групповой. В частном случае, если групповой адрес состоит из всех единиц, то есть имеет шестнадцатеричное представление 0xFFFFFFFFFFFF, он идентифицирует все узлы сети и называется **широковещательным**.

Второй бит старшего байта адреса определяет способ назначения адреса — **централизованный** или **локальный**. Если этот бит равен 0 (что бывает почти всегда в стандартной аппаратуре Ethernet), это говорит о том, что адрес назначен централизованно по правилам IEEE 802.

ВНИМАНИЕ

В стандартах IEEE Ethernet младший бит байта изображается в самой левой позиции поля, а старший бит — в самой правой. Этот нестандартный способ отображения порядка следования битов в байте соответствует порядку передачи битов в линию связи передатчиком Ethernet (первым передается младший бит). В стандартах других организаций, например RFC IETF, ITU-T, ISO, используется традиционное представление байта, когда младший бит считается самым правым битом байта, а старший — самым левым. При этом порядок следования байтов остается традиционным. Поэтому при чтении стандартов, опубликованных этими организациями, а также чтении данных, отображаемых на экране операционной системой или анализатором протоколов, значения каждого байта кадра Ethernet нужно зеркально отобразить, чтобы получить представление о значении разрядов этого байта в соответствии с документами IEEE. Например, групповой адрес, имеющий в нотации IEEE вид 1000 0000 0000 0000 1010 0111 1111 0000 0000 0000 0000 0000 или в шестнадцатеричной записи 80-00-A7-F0-00-00, будет, скорее всего, отображен анализатором протоколов в традиционном виде как 01-00-E5-0F-00-00.

Комитет IEEE распределяет между производителями оборудования так называемые **организационно уникальные идентификаторы** (Organizationally Unique Identifier, OUI). Каждый производитель помещает выделенный ему идентификатор в три старших байта адреса (например, идентификатор 0x0020AF определяет компанию 3COM, а 0x00000C – Cisco). За уникальность младших трех байтов адреса отвечает производитель оборудо-

вания. Двадцать четыре бита, отводимые производителю для адресации интерфейсов его продукции, позволяют выпустить примерно 16 миллионов интерфейсов под одним идентификатором организации. Уникальность централизованно распределяемых адресов распространяется на все основные технологии локальных сетей — Ethernet, Token Ring, FDDI и т. д. Локальные адреса назначаются администратором сети, в обязанности которого входит обеспечение их уникальности.

Сетевые адаптеры Ethernet могут также работать в так называемом **режиме неразборчивого захвата** (promiscuous mode), когда они захватывают все кадры, поступающие на интерфейс, независимо от их MAC-адресов назначения. Обычно такой режим используется для мониторинга трафика, когда захваченные кадры изучаются затем для нахождения причины некорректного поведения некоторого узла или отладки нового протокола.

Форматы кадров технологии Ethernet

Существует несколько стандартов формата кадра Ethernet. На практике в оборудовании Ethernet используется только один формат кадра, а именно кадр Ethernet DIX, который иногда называют кадром Ethernet II по номеру последнего стандарта DIX. Этот формат представлен на рис. 12.5.

6 байт	6 байт	2 байта	46–1500 байт	4 байта
DA	SA	T	Данные	FCS

Рис. 12.5. Формат кадра Ethernet DIX (II)

Первые два поля заголовка отведены под адреса:

- **DA (Destination Address)** — MAC-адрес узла назначения;
- **SA (Source Address)** — MAC-адрес узла отправителя.

  Для доставки кадра достаточно одного адреса — адреса назначения; адрес источника помещается в кадр для того, чтобы узел, получивший кадр, знал, от кого пришел кадр и кому нужно на него ответить. Принятие решения об ответе не входит в компетенцию протокола Ethernet, это дело протоколов верхних уровней. Ethernet же только выполнит такое действие, если с сетевого уровня поступит соответствующее указание.

- **Поле T (Type**, или **EtherType)** содержит условный код протокола верхнего уровня, данные которого находятся в поле данных кадра, например шестнадцатеричное значение 08-00 соответствует протоколу IP. Это поле требуется для поддержки интерфейсных функций мультиплексирования и демультиплексирования кадров при взаимодействии с протоколами верхних уровней.
- **Поле данных** может содержать от 46 до 1500 байт. Если длина пользовательских данных меньше 46 байт, то это поле дополняется до минимального размера байтами заполнения. Эта операция требуется для корректной работы метода доступа Ethernet (он рассматривается в следующем разделе).
- **Поле контрольной последовательности кадра** (Frame Check Sequence, FCS) состоит из 4 байт контрольной суммы. Это значение вычисляется по алгоритму CRC-32.

Кадр Ethernet DIX (II) не отражает разделения канального уровня Ethernet на уровень MAC и уровень LLC: его поля поддерживают функции обоих уровней, например интер-

фейсные функции поля *T* относятся к функциям уровня LLC, в то время как все остальные поля поддерживают функции уровня MAC.

Существуют еще три стандартных формата кадра Ethernet:

- Кадр 802.3/LLC является стандартом комитета IEEE 802 и построен в соответствии с принятым разбиением функций канального уровня на уровень MAC и уровень LLC. Поэтому результирующий кадр является вложением кадра LLC, определяемого стандартом 802.2, в кадр MAC, определяемого стандартом 802.3.
- Кадр **Raw 802.3**, или **Novell 802.3**, появился в результате усилий компании Novell по ускорению разработки своего стека протоколов в сетях Ethernet.
- Кадр **Ethernet SNAP** стал результатом деятельности комитета 802.2 по приведению предыдущих форматов кадров к некоторому общему стандарту и приданию кадру необходимой гибкости для учета в будущем возможностей добавления полей или изменения их назначения.

Как уже было сказано, в настоящее время оборудованием Ethernet используются только кадры Ethernet DIX (II). Остальные форматы кадров, в том числе кадр 802.3/LLC, по-прежнему формально являющийся стандартным, вышли из употребления из-за более сложного формата, который оказался не нужен в условиях существования единой технологии канального уровня.

Более подробную информацию о форматах кадров Ethernet можно найти на сайте www.olifer.co.uk в документе «Форматы кадров Ethernet».

Доступ к среде и передача данных

Метод доступа, используемый в сетях Ethernet на разделяемой проводной среде[1], носит название CSMA/CD (Carrier Sense Multiple Access with Collision Detection — прослушивание несущей частоты с множественным доступом и распознаванием коллизий). Название метода достаточно хорошо описывает его особенности.

Все компьютеры в сети на разделяемой среде имеют возможность немедленно (с учетом задержки распространения сигнала в физической среде) получить данные, которые любой из компьютеров начал передавать в общую среду. Говорят, что среда, к которой подключены все станции, работает в режиме **коллективного доступа** (Multiply Access, MA).

Чтобы получить возможность передавать кадр, интерфейс-отправитель должен убедиться, что разделяемая среда свободна. Это достигается прослушиванием основной гармоники сигнала, которая еще называется **несущей частотой** (Carrier Sense, CS).

Признаком «незанятости» среды является отсутствие на ней несущей частоты, которая при манчестерском способе кодирования равна 5–10 МГц в зависимости от последовательности единиц и нулей, передаваемых в данный момент.

---

[1] В беспроводных сетях Ethernet применяется другой метод доступа, известный как CSMA/CA. Этот метод рассматривается далее в разделе «Беспроводные локальные сети IEEE 802.11».

Если среда свободна, то узел имеет право начать передачу кадра. В примере, показанном на рис. 12.6, узел 1 обнаружил, что среда свободна, и начал передавать свой кадр. В коаксиальном кабеле сигналы передатчика узла 1 распространяются в обе стороны, так что их получают все узлы сети. Кадр данных всегда сопровождается **преамбулой**, которая состоит из 7 байт, каждый из которых имеет значение 10101010, и 8-го байта, равного 10101011. Последний байт носит название **ограничителя начала кадра**. Преамбула нужна для вхождения приемника в побитовую и побайтовую синхронизацию с передатчиком. Наличие двух единиц, идущих подряд, говорит приемнику о том, что преамбула закончилась и следующий бит является началом кадра.

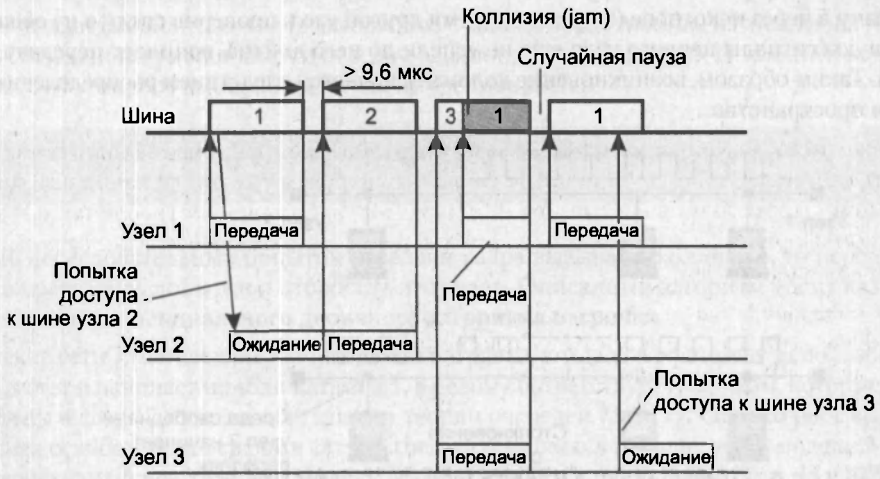


Рис. 12.6. Метод случайного доступа CSMA/CD

Все станции, подключенные к кабелю, начинают записывать байты передаваемого кадра в свои внутренние буферы. Первые 6 байт кадра содержат адрес назначения. Та станция, которая узнает собственный адрес в заголовке кадра, продолжает записывать его содержимое в свой внутренний буфер, а остальные станции на этом прием кадра прекращают. Станция назначения обрабатывает полученные данные и передает их вверх по своему стеку. Кадр Ethernet содержит не только адрес назначения, но и адрес источника данных, поэтому станция-получатель знает, кому нужно послать ответ.

Узел 2 во время передачи кадра узлом 1 также пытался начать передачу своего кадра, однако обнаруживает, что среда занята — на ней присутствует несущая частота, — поэтому узел 2 вынужден ждать, пока узел 1 не прекратит передачу кадра.

После окончания передачи кадра все узлы сети обязаны выдержать технологическую паузу, равную **межпакетному интервалу** (Inter Packet Gap, IPG) в 9,6 мкс. Эта пауза нужна для приведения сетевых адаптеров в исходное состояние, а также для предотвращения монопольного захвата среды одной станцией. После окончания технологической паузы узлы имеют право начать передачу своего кадра, так как среда свободна. В приведенном примере узел 2 дождался окончания передачи кадра узлом 1, сделал паузу в 9,6 мкс и начал передачу своего кадра.

Возникновение коллизии

Механизм прослушивания среды и пауза между кадрами не гарантируют исключения ситуации, когда две или более станции одновременно решают, что среда свободна, и начинают передавать свои кадры. Говорят, что при этом происходит **коллизия**, так как содержимое обоих кадров сталкивается на общем кабеле и происходит искажение информации.

Коллизия — это нормальная ситуация в работе сетей Ethernet. В примере на рис. 12.7 коллизию породила одновременная передача данных узлами 3 и 1. Для возникновения коллизии не обязательно, чтобы несколько станций начали передачу *абсолютно* одновременно, такая ситуация маловероятна. Более вероятна ситуация, когда один узел начинает передачу, а через некоторое (короткое) время другой узел, проверив среду и не обнаружив несущую (сигналы первого узла еще не успели до него дойти), начинает передачу своего кадра. Таким образом, возникновение коллизии является следствием распределения узлов сети в пространстве.

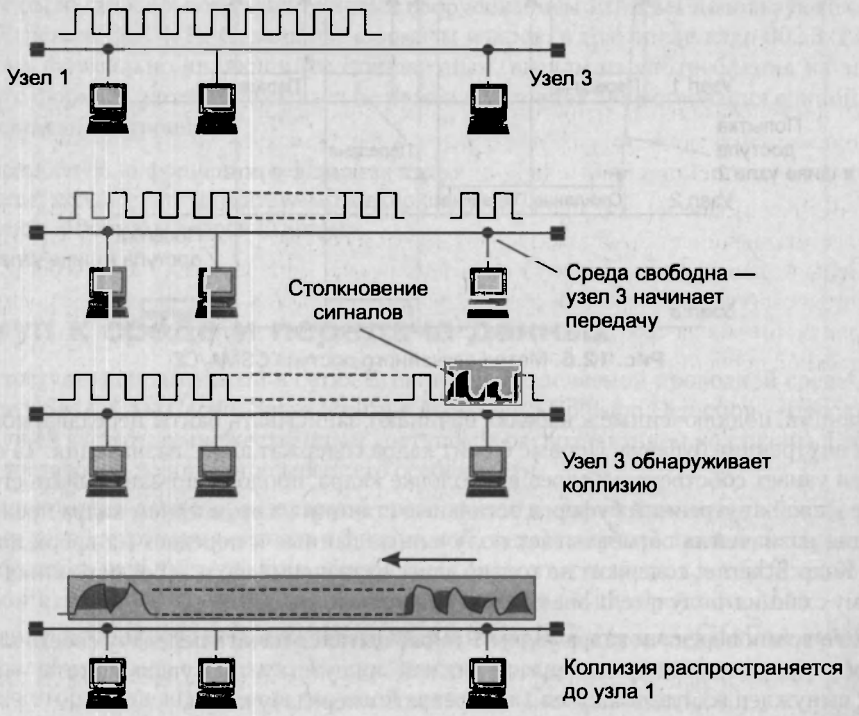


Рис. 12.7. Схема возникновения и распространения коллизии

Чтобы корректно обработать коллизию, все станции одновременно наблюдают за возникающими на кабеле сигналами. Если передаваемые и наблюдаемые сигналы отличаются, то фиксируется факт **обнаружения коллизии** (Collision Detection, CD). Для повышения вероятности скорейшего обнаружения коллизии всеми станциями сети станция, которая обнаружила коллизию, прерывает передачу своего кадра (в произвольном месте, возможно, и не на границе байта) и усугубляет коллизию посылкой в сеть специальной последовательности из 32 бит, называемой **jam-последовательностью**.

После этого обнаружившая коллизию передающая станция обязана прекратить передачу и сделать паузу в течение короткого случайного интервала времени. Затем она может снова предпринять попытку захвата среды и передачи кадра. Случайная пауза выбирается по следующему алгоритму:

$$\text{Пауза} = L \times (\text{интервал отсрочки}).$$

В технологии Ethernet **интервал отсрочки** выбран равным значению 512 битовых интервалов. Битовый интервал соответствует времени между появлением двух последовательных битов данных на кабеле; для скорости 10 Мбит/с величина битового интервала равна 0,1 мкс, или 100 нс.

L представляет собой целое число, выбранное с равной вероятностью из диапазона $[0, 2N]$, где N – номер повторной попытки передачи данного кадра: 1, 2, ..., 10. После 10-й попытки интервал, из которого выбирается пауза, не увеличивается.

Таким образом, случайная пауза в технологии Ethernet может принимать значения от 0 до 52,4 мс.

Если 16 последовательных попыток передачи кадра вызывают коллизию, то передатчик должен прекратить попытки и отбросить этот кадр. Описанный алгоритм носит название **усеченного экспоненциального двоичного алгоритма отсрочки**.

Поведение сети Ethernet при значительной нагрузке, когда коэффициент использования среды растет и начинает приближаться к 1, в целом соответствует графикам, которые были приведены в главе 7 при анализе модели теории очередей M/M/1. Однако рост времени ожидания освобождения среды в сетях Ethernet начинается раньше, чем в модели M/M1. Это происходит из-за того, что модель M/M/1 является очень простой и не учитывает такой важной особенности Ethernet, как коллизии.

Администраторы сетей Ethernet на разделяемой среде руководствуются простым эмпирическим правилом – коэффициент использования среды не должен превышать 30 %. Для поддержки чувствительного к задержкам трафика сети Ethernet (и другие сети на разделяемой среде) могут применять только один метод поддержания характеристик QoS – *недогруженный режим работы*.

Время оборота и распознавание коллизий

Надежное распознавание коллизий всеми станциями сети является необходимым условием корректной работы сети Ethernet. Если какая-либо передающая станция не распознает коллизию и решит, что кадр данных передан ею верно, этот кадр будет утерян. Из-за наложения сигналов при коллизии информация кадра исказится, и он будет отбракован принимающей станцией из-за несовпадения контрольной суммы. Скорее всего, недошедшие до получателя данные будут повторно переданы каким-либо протоколом верхнего уровня, например транспортным или прикладным, работающим с установлением соединения, либо протоколом LLC, если он работает в режиме LLC2. Однако повторная передача сообщения протоколами верхних уровней произойдет гораздо позже (иногда по прошествии нескольких секунд), чем повторная передача средствами сети Ethernet, работающей с микросекундными интервалами. Поэтому если коллизии не будут надежно

распознаваться узлами сети Ethernet, то это приведет к заметному снижению полезной пропускной способности сети.

Для надежного распознавания коллизий должно выполняться следующее соотношение:

$$T_{min} > RTT.$$

Здесь T_{min} — время передачи кадра минимальной длины, а RTT — *время оборота*, то есть время, за которое сигнал коллизии успевает распространиться до самого дальнего узла сети. В худшем случае сигнал должен пройти дважды между наиболее удаленными друг от друга станциями сети (в одну сторону проходит неискаженный сигнал, а в обратном направлении — сигнал, уже искаженный коллизией).

При выполнении этого условия передающая станция должна успеть обнаружить коллизию, которую вызвал переданный ее кадр, еще до того, как она закончит передачу этого кадра.

Очевидно, что выполнение этого условия зависит, с одной стороны, от минимальной длины кадра и скорости передачи данных протокола, а с другой стороны, от длины кабельной системы сети и скорости распространения сигнала в кабеле (для разных типов кабеля эта скорость несколько отличается).

Все параметры протокола Ethernet подобраны таким образом, чтобы при нормальной работе сети коллизии четко распознавались.

Так, стандарт Ethernet определяет минимальную длину поля данных кадра в 46 байт (что вместе со служебными полями дает минимальную длину кадра 64 байт, а вместе с преамбулой — 72 байт, или 576 бит). Отсюда может быть вычислено ограничение на расстояние между станциями. В стандарте Ethernet 10 Мбит/с время передачи кадра минимальной длины равно 575 битовых интервалов, следовательно, время оборота должно быть меньше 57,5 мкс. Расстояние, которое сигнал может пройти за это время, зависит от типа кабеля и для толстого коаксиального кабеля равно примерно 13 280 м. Учитывая, что за время 57,5 мкс сигнал должен пройти по линии связи дважды, расстояние между двумя узлами не должно быть больше 6635 м. В стандарте величина этого расстояния выбрана равной 2500 м, что существенно меньше. Это объясняется тем, что повторители, которые нужны для соединения пяти сегментов кабеля, вносят задержки в распространение сигнала.

Описанные соображения объясняют выбор минимальной длины поля данных кадра в 46 байт. Уменьшение этого значения до 0 привело бы к значительному сокращению максимальной длины сети.

Требование $T_{min} > RTT$ имеет одно интересное следствие: чем выше скорость протокола, тем меньше должна быть максимальная длина сети. Поэтому для Ethernet на разделяемой среде при скорости в 100 Мбит/с максимальная длина сети пропорционально уменьшается до 250 м, а при скорости в 1 Гбит/с — до 25 м. Эта зависимость, наряду с резким ростом задержек при повышении загрузки сети, говорит о еще одном коренном недостатке метода доступа CSMA/CD.

Спецификации физической среды

При стандартизации технологии Ethernet рабочей группой IEEE 802.3 вариант Ethernet на «толстом» коаксиальном кабеле получил название 10Base-5.

Число 10 этом названии обозначает номинальную битовую скорость передачи данных стандарта, то есть 10 Мбит/с, а слово «Base» — метод передачи на одной базовой частоте[1] (в данном случае 10 МГц). Последний символ в названии стандарта физического уровня обозначает тип кабеля, в данном случае 5 отражает тот факт, что диаметр «толстого» коаксиала равен 0,5 дюйма. Данная система обозначения типа физического уровня Ethernet сохранилась до настоящего времени.

Наиболее популярными спецификациями физической среды Ethernet для скорости передачи данных 10 Мбит/с являются следующие:

- **10Base-5** — коаксиальный кабель диаметром 0,5 дюйма, называемый «толстым» коаксиалом. Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента: 500 м (без повторителей). Максимальное количество узлов подключаемых к сегменту — 100. Максимальное число сегментов — 5 (4 повторителя), из которых только 3 могут использоваться для подключения узлов, а 2 играют роль удлинителей сети.
- **10Base-2** — коаксиальный кабель диаметром 0,25 дюйма, называемый «тонким» коаксиалом. Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента — 185 м (без повторителей). Максимальное количество узлов подключаемых к сегменту — 30. Максимальное число сегментов — 5 (4 повторителя), из которых только 3 могут использоваться для подключения узлов, а 2 играют роль удлинителей сети.
- **10Base-T** — кабель на основе неэкранированной витой пары (UTP). Образует звездообразную топологию на основе концентратора (многопортового повторителя). Расстояние между концентратором и конечным узлом — не более 100 м. Между любыми двумя узлами сети может быть не более 4-х концентраторов (так называемое «правило 4-х хабов»).
- **10Base-F** — волоконно-оптический кабель. Топология аналогична топологии стандарта 10Base-T, но расстояние между концентратором и конечным узлом может достигать 2000 м. Правило 4-х хабов остается в силе.

В *стандарте 10Base-2* в качестве передающей среды используется «тонкий» коаксиал Ethernet. Тонкий коаксиальный кабель дешевле толстого, поэтому сети 10Base-2 иногда называли Cheapernet (дословно — дешевая сеть). Станции подключаются к кабелю с помощью высокочастотного **Т-коннектора**, представляющего собой тройник, один отвод которого соединяется с сетевым адаптером, а два других — с двумя концами разрыва кабеля. Стандарт 10Base-2 очень близок к стандарту 10Base-5, но трансиверы в нем объединены с сетевыми адаптерами за счет того, что более гибкий тонкий коаксиальный кабель может быть подведен непосредственно к выходному разъему платы сетевого адаптера, установленной в шасси компьютера. Кабель в данном случае «висит» на сетевом адаптере, что затрудняет физическое перемещение компьютеров, однако сама операция соединения компьютеров в сеть оказывается гораздо проще, чем для сети на «толстом» коаксиале.

Реализация этого стандарта на практике приводит к наиболее простому решению для кабельной сети, так как для соединения компьютеров требуются только сетевые адаптеры, Т-коннекторы и терминаторы на 50 Ом. Однако этот вид кабельных соединений наиболее сильно подвержен авариям и сбоям. Кабель более восприимчив к помехам, чем «толстый»

---

[1] В отличие от методов, использующих несколько несущих частот; такие методы называются широкополосными и имеют в своем составе слово «Broadband». Эти методы, хотя и были стандартизованы, не получили распространения в период популярности локальных сетей на разделяемой среде.

коаксиал. В моноканале имеется большое количество механических соединений: каждый Т-коннектор дает три механических соединения, два из которых имеют жизненное значение для всей сети. Пользователи имеют доступ к разъемам и могут нарушить целостность моноканала. Кроме того, эстетика и эргономичность этого решения оставляют желать лучшего, так как от каждой станции через Т-коннектор отходят два довольно заметных провода, которые под столом часто образуют моток кабеля — запас, необходимый на случай даже небольшого перемещения рабочего места.

Сеть Eternet на витой паре, описываемая стандартом 10Base-T, стала следующим шагом на пути повышения эксплуатационных характеристик Ethernet.

Одним из существенных недостатков Ethernet на коаксиальном кабеле являлось отсутствие оперативной информации о состоянии кабеля и сложность нахождения места его повреждения. Поэтому поиск неисправностей стал привычной процедурой и головной болью многочисленной армии сетевых администраторов коаксиальных сетей Ethernet.

Альтернатива появилась в середине 80-х годов, когда благодаря использованию витой пары и повторителей сети Ethernet стали гораздо более ремонтопригодными.

К этому времени телефонные компании уже достаточно давно применяли многопарный кабель на основе неэкранированной витой пары для подключения телефонных аппаратов внутри зданий. Идея приспособить этот популярный вид кабеля для локальных сетей оказалась очень плодотворной, так как многие здания уже были оснащены нужной кабельной системой. Оставалось разработать способ подключения сетевых адаптеров и прочего коммуникационного оборудования к витой паре таким образом, чтобы изменения в сетевых адаптерах и программном обеспечении сетевых операционных систем были минимальными по сравнению с сетями Ethernet на коаксиале. Эта попытка оказалась успешной — переход на витую пару требует только замены приемника и передатчика сетевого адаптера, а метод доступа и все протоколы канального уровня остаются теми же, что и в сетях Ethernet на коаксиале.

Правда, для соединения узлов в сеть теперь обязательно требуется коммуникационное устройство — **многопортовый повторитель** Ethernet на витой паре.

Устройство такого повторителя схематично изображено на рис. 12.8. Каждый сетевой адаптер соединяется с повторителем двумя витыми парами. Одна витая пара требуется для передачи данных от станции к повторителю (выход *TX* сетевого адаптера), другая — для передачи данных от повторителя к станции (вход *RX* сетевого адаптера). Повторитель побитно принимает сигналы от одного из конечных узлов и синхронно передает их на все свои остальные порты, исключая тот, с которого поступили сигналы, одновременно улучшая их электрические характеристики.

Многопортовый повторитель часто называют **концентратором**, или **хабом** (от английского hub — центр, ступица колеса), так как в нем сконцентрированы соединения со всеми конечными узлами сети. Фактически хаб имитирует сеть на коаксиальном кабеле в том отношении, что физически отдельные отрезки кабеля на витой паре логически все равно представляют единую разделяемую среду. Все правила доступа к среде по алгоритму CSMA/CD сохраняются.

При создании сети Ethernet на витой паре с большим числом конечных узлов хабы можно соединять друг с другом иерархическим способом, образуя *древовидную структуру* (рис. 12.9). Добавление каждого хаба изменяет физическую структуру, но оставляет без изменения логическую структуру сети. То есть независимо от числа хабов в сети сохраня-

ется одна общая для всех интерфейсов разделяемая среда, так что передача кадра с любого интерфейса блокирует передатчики всех остальных интерфейсов.

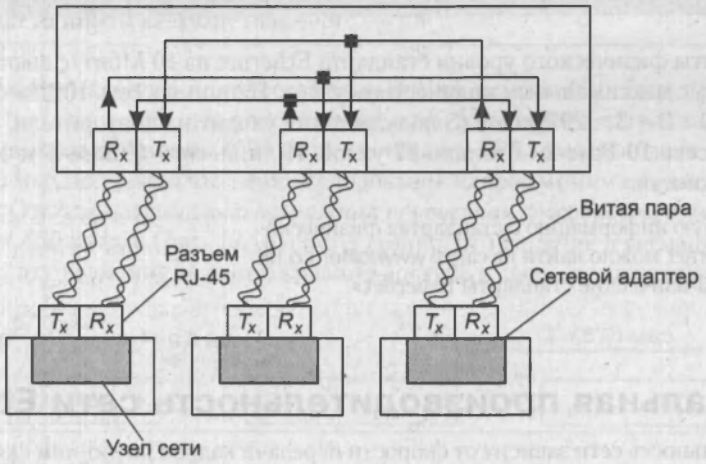


Рис. 12.8. Повторитель Ethernet на витой паре

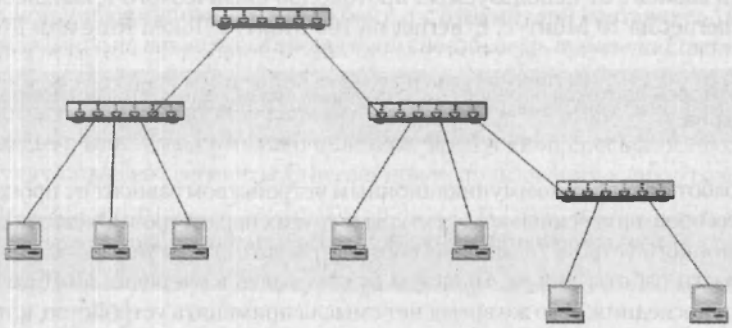


Рис. 12.9. Иерархическое соединение хабов

Физическая структуризация сетей, построенных на основе витой пары, повышает надежность и упрощает обслуживание сети, поскольку в этом случае появляется возможность контролировать состояние и локализовывать отказы отдельных кабельных отрезков, подключающих конечные узлы к концентраторам. В случае обрыва, короткого замыкания или неисправности сетевого адаптера работа сети может быть быстро восстановлена путем отключения соответствующего сегмента кабеля.

Для контроля целостности физического соединения между двумя непосредственно соединенными портами в стандарте 10Base-T введен так называемый **тест целостности соединения** (Link Integrity Test, LIT). Эта процедура заключается в том, что в те периоды, когда порт не посылает или получает кадры данных, он посылает своему соседу импульсы длительностью 100 нс через каждые 16 мс. Если порт принимает такие импульсы от своего соседа, то он считает соединение работоспособным и, как правило, индицирует это зеленым светом светодиода.

Независимо от используемого физического уровня в стандартах Ethernet на 10 Мбит/с вводится ограничение на максимальное количество узлов, подключаемых к разделяемой среде. Это ограничение составляет 1024 узла.

Не все варианты физического уровня стандарта Ethernet на 10 Мбит/с дают возможность построить сеть с максимальным количеством узлов. Например, сеть 10Base-5 может иметь максимум 100 × 3 – 3 = 297 узлов (3 подключения уходят на повторители, соединяющие сегменты), а сеть 10 Base-2 — только 87 узлов. И лишь сети 10Base-T и 10Base-F дают такую возможность.

Более подробную информацию о стандартах физического уровня Ethernet можно найти на сайте www.olifer.co.uk в документе «Физические стандарты Ethernet».

Максимальная производительность сети Ethernet

Производительность сети зависит от скорости передачи кадров по линиям связи и скорости обработки этих кадров коммуникационными устройствами, передающими кадры между своими портами, к которым эти линии связи подключены. Скорость передачи кадров по линиям связи зависит от используемых протоколов физического и канального уровней, например Ethernet на 10 Мбит/с, Ethernet на 100 Мбит/с, Token Ring или FDDI.

Скорость, с которой протокол передает биты по линии связи, называется **номинальной скоростью протокола**.

Скорость обработки кадров коммуникационным устройством зависит от производительности его процессоров, внутренней архитектуры и других параметров. Очевидно, что скорость коммуникационного устройства должна соответствовать скорости работы линии. Если она меньше скорости работы линии, то кадры будут стоять в очередях и отбрасываться при переполнении последних. В то же время нет смысла применять устройство, которое в сотни раз производительнее, чем того требует скорость подключаемых к нему линий.

Для оценки требуемой производительности коммуникационных устройств, имеющих порты Ethernet, необходимо оценить производительность *сегмента Ethernet*, но не в битах в секунду (ее мы знаем — это 10 Мбит/с), а в кадрах в секунду, так как именно этот показатель помогает оценить требования к производительности коммуникационных устройств. Это объясняется тем, что на обработку каждого кадра, независимо от его длины, мост, коммутатор или маршрутизатор тратит примерно равное время, которое уходит на просмотр таблицы продвижения пакета, формирование нового кадра (для маршрутизатора) и т. п.

При постоянной битовой скорости количество кадров, поступающих на коммуникационное устройство в единицу времени, является, естественно, максимальным при их минимальной длине. Поэтому для коммуникационного оборудования наиболее тяжелым режимом является обработка потока кадров *минимальной длины*.

Теперь рассчитаем максимальную производительность сегмента Ethernet в таких единицах, как число переданных кадров (пакетов) минимальной длины в секунду.

ПРИМЕЧАНИЕ

При указании производительности сетей термины «кадр» и «пакет» обычно используются как синонимы. Соответственно, аналогичными являются и единицы измерения производительности кадры в секунду (кадр/с) и пакеты в секунду (пакет/с).

Для расчета максимального количества кадров минимальной длины, проходящих по сегменту Ethernet, вспомним, что подсчитанное нами ранее время, затрачиваемое на передачу кадра минимальной длины (576 бит), составляет 57,5 мкс. Прибавив межкадровый интервал в 9,6 мкс, получаем, что период следования кадров минимальной длины составляет 67,1 мкс. Отсюда *максимально возможная пропускная способность сегмента Ethernet составляет 14 880 кадр/с* (рис. 12.10). Естественно, что наличие в сегменте нескольких узлов снижает эту величину за счет ожидания доступа к среде, а также за счет коллизий.

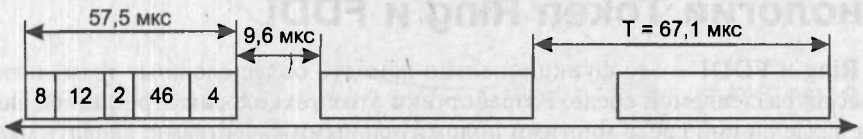


Рис. 12.10. К расчету пропускной способности протокола Ethernet

Кадры максимальной длины технологии Ethernet имеют поле данных 1500 байт, что вместе со служебной информацией дает 1518 байт, а с преамбулой составляет 1526 байт, или 12 208 бит. *Максимально возможная пропускная способность сегмента Ethernet для кадров максимальной длины составляет 813 кадр/с.* Очевидно, что при работе с большими кадрами нагрузка на мосты, коммутаторы и маршрутизаторы довольно ощутимо снижается.

Теперь рассчитаем, какой максимально полезной пропускной способностью, измеряемой в битах в секунду, обладают сегменты Ethernet при использовании кадров разного размера.

Полезной пропускной способностью протокола называется максимальная скорость передачи *пользовательских* данных, которые переносятся полем данных кадра.

Эта пропускная способность всегда меньше номинальной битовой скорости протокола Ethernet за счет нескольких факторов:

- служебной информации кадра;
- межкадровых интервалов (IPG);
- ожидания доступа к среде.

Для кадров минимальной длины полезная пропускная способность равна:

$$B = 14880 \times 46 \times 8 = 5{,}48 \text{ Мбит/с}.$$

Это несколько меньше, чем 10 Мбит/с, но следует учесть, что кадры минимальной длины используются в основном для передачи квитанций, так что к передаче собственно данных файлов эта скорость имеет небольшое отношение.

Для кадров максимальной длины полезная пропускная способность равна:

$$B_{\text{п}} = 813 \times 1500 \times 8 = 9{,}76 \text{ Мбит/с}.$$

При использовании кадров среднего размера с полем данных в 512 байт пропускная способность протокола составляет 9,29 Мбит/с.

В двух последних случаях пропускная способность протокола оказалась достаточно близкой к предельной пропускной способности в 10 Мбит/с, однако следует учесть, что при расчете мы предполагали, что двум взаимодействующим станциям «не мешают» никакие другие станции сети, то есть отсутствуют коллизии и ожидание доступа.

Таким образом, при отсутствии коллизий коэффициент использования сети зависит от размера поля данных кадра и имеет максимальное значение 0,976 при передаче кадров максимальной длины.

Технологии Token Ring и FDDI

Token Ring и **FDDI** — это функционально намного более сложные технологии, чем Ethernet на разделяемой среде. Разработчики этих технологий стремились наделить сеть на разделяемой среде многими положительными качествами: сделать механизм разделения среды предсказуемым и управляемым, обеспечить отказоустойчивость сети, организовать приоритетное обслуживание для чувствительного к задержкам трафика, например голосового. Нужно отдать им должное — во многом их усилия оправдались, и сети FDDI довольно долгое время успешно использовались как магистрали сетей масштаба кампуса, в особенности в тех случаях, когда нужно было обеспечить высокую надежность магистрали.

Механизм доступа к среде в сетях Token Ring и FDDI является более детерминированным, чем в сетях Ethernet.

Рассмотрим его на примере сети **Token Ring**, станции которой связаны в кольцо (рис. 12.11), так что любая станция непосредственно получает данные только от одной станции — той, которая является предыдущей в кольце, а передает данные своему ближайшему соседу вниз по потоку данных. Скорость передачи данных в первых сетях Token Ring, разработанных компанией IBM, была всего 4 Мбит/с, но затем была повышена до 16 Мбит/с. Основная среда передачи данных — витая пара. Для адресации станций сети Token Ring (и FDDI) используют MAC-адреса того же формата, что и Ethernet.

Метод доступа Token Ring основан на передаче от узла к узлу специального кадра — **токена**, или **маркера доступа**, при этом только узел, владеющий токеном, может передавать свои кадры в кольцо, которое становится в этом случае разделяемой средой. Существует лимит на период монопольного использования среды — это так называемое **время удержания токена**, по истечение которого станция обязана передать токен своему соседу по кольцу. В результате такие ситуации, как неопределенное время ожидания доступа к среде, характерные для Ethernet, здесь исключены (по крайней мере, в тех случаях, когда сетевые адаптеры станций исправны и работают без сбоев). Максимальное время ожидания всегда нетрудно оценить, так как оно равно произведению времени удержания токена на количество станций в кольце. Так как станция, получившая токен, но не имеющая в этот момент кадров для передачи, передает токен следующей станции, то время ожидания может быть меньше.

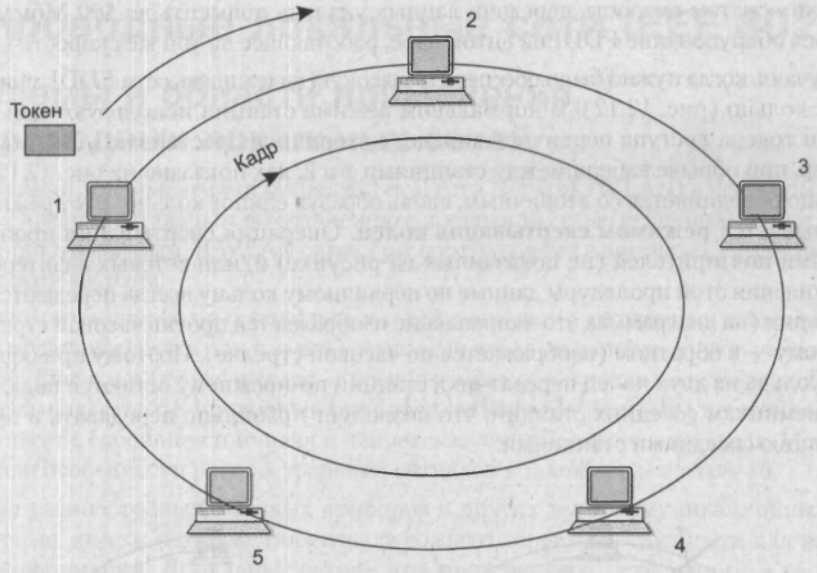


Рис. 12.11. Сеть Token Ring

Отказоустойчивость сети Token Ring определяется использованием в сети повторителей (не показанных на рис. 12.11) для создания кольца. Каждый такой повторитель имеет несколько портов, которые образуют кольцо за счет внутренних связей между передатчиками и приемниками. В случае отказа или отсоединения станции повторитель организует обход порта этой станции, так что связность кольца не нарушается.

Поддержка чувствительного к задержкам трафика достигается за счет *системы приоритетов кадров*. Решение о приоритете конкретного кадра принимает передающая станция. Токен также всегда имеет некоторый уровень текущего приоритета. Станция имеет право захватить переданный ей токен только в том случае, если приоритет кадра, который она хочет передать, выше приоритета токена (или равен ему). В противном случае станция обязана передать токен следующей по кольцу станции.

Благодаря более высокой, чем в сетях Ethernet, скорости, детерминированности распределения пропускной способности сети между узлами, а также лучших эксплуатационных характеристик (обнаружение и изоляция неисправностей), сети Token Ring были предпочтительным выбором для таких чувствительных к подобным показателям приложений, как банковские системы и системы управления предприятием.

Технологию **FDDI** можно считать усовершенствованным вариантом Token Ring, так как в ней, как и в Token Ring, используется метод доступа к среде, основанный на передаче токена, а также кольцевая топология связей, но вместе с тем FDDI работает на более высокой скорости и имеет более совершенный механизм отказоустойчивости.

Технология FDDI стала первой технологией локальных сетей, в которой оптическое волокно, начавшее применяться в телекоммуникационных сетях с 70-х годов прошлого века, было использовано в качестве разделяемой среды передачи данных. За счет применения

оптических систем скорость передачи данных удалось повысить до 100 Мбит/с (позже появилось оборудование FDDI на витой паре, работающее на той же скорости).

В тех случаях, когда нужно было обеспечить высокую надежность сети FDDI, применялось двойное кольцо (рис. 12.12). В нормальном режиме станции используют для передачи данных и токена доступа первичное кольцо, а вторичное простаивает[1]. В случае отказа, например, при обрыве кабеля между станциями 1 и 2, как показано на рис. 12.12, первичное кольцо объединяется со вторичным, вновь образуя единое кольцо. Этот режим работы сети называется **режимом свертывания колец**. Операция свертывания производится средствами повторителей (не показанных на рисунке) и/или сетевых адаптеров FDDI. Для упрощения этой процедуры данные по первичному кольцу всегда передаются в одном направлении (на диаграммах это направление изображается против часовой стрелки), а по вторичному — в обратном (изображается по часовой стрелке). Поэтому при образовании общего кольца из двух колец передатчики станций по-прежнему остаются подключенными к приемникам соседних станций, что позволяет правильно передавать и принимать информацию соседними станциями.

Рис. 12.12. Отказоустойчивость в сети FDDI

В стандартах FDDI много внимания отводится различным процедурам, которые позволяют определить факт наличия отказа в сети, а затем произвести необходимое реконфигурирование. Технология FDDI расширяет механизмы обнаружения отказов технологии Token Ring за счет резервных связей, которые предоставляет второе кольцо.

Более подробную информация о технологиях Token Ring и FDDI можно найти на сайте www.olifer.co.uk в документах «Технология Token Ring» и «Технология FDDI».

[1] Существовали фирменные реализации оборудования FDDI, в которых в нормальном режиме использовалось и вторичное кольцо. Тем самым удавалось добиваться удвоения скорости передачи данных.

Беспроводные локальные сети IEEE 802.11

Проблемы и области применения беспроводных локальных сетей

Беспроводные локальные сети (Wireless Local Area Network, WLAN) в некоторых случаях являются предпочтительным по сравнению с проводной сетью решением, а иногда просто единственно возможным. В WLAN сигнал распространяется с помощью электромагнитных волн высокой частоты.

Преимущество беспроводных локальных сетей очевидно — их проще и дешевле разворачивать и модифицировать, так как вся громоздкая кабельная инфраструктура оказывается излишней. Еще одно преимущество — обеспечение мобильности пользователей. Однако за эти преимущества беспроводные сети расплачиваются длинным перечнем проблем, которые несет с собой неустойчивая и непредсказуемая беспроводная среда. Мы уже рассматривали особенности распространения сигналов в такой среде в главе 10.

Помехи от разнообразных бытовых приборов и других телекоммуникационных систем, атмосферные помехи и отражения сигнала создают серьезные трудности для надежного приема информации. Локальные сети — это, прежде всего, сети зданий, а распространение радиосигнала внутри здания еще сложнее, чем вне его. В стандарте IEEE 802.11 приводится изображение распределения интенсивности сигнала (рис. 12.13). В стандарте подчеркивается, что это статическое изображение, в действительности картина является динамической, и при перемещении объектов в комнате распределение сигнала может существенно измениться.

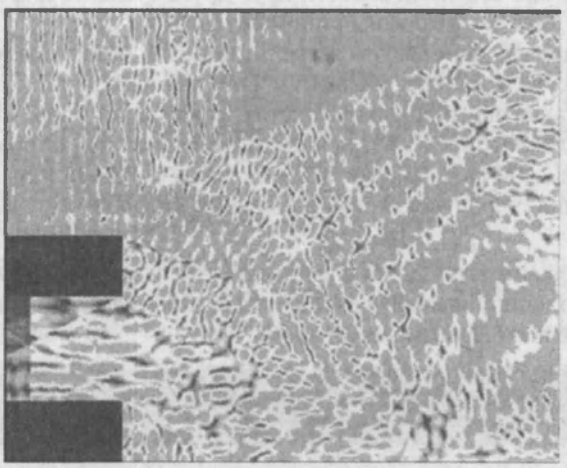


Рис. 12.13. Распределение интенсивности радиосигнала

Методы *расширения спектра* помогают снизить влияние помех на полезный сигнал, кроме того, в беспроводных сетях широко используются *прямая коррекция ошибок* (FEC) и протоколы с повторной передачей потерянных кадров. Тем не менее практика показала, что в тех случаях, когда ничего не мешает применению проводной локальной сети, органи-

зации предпочитают именно этот вид LAN, несмотря на то что при этом нельзя обойтись без кабельной системы.

Неравномерное распределение интенсивности сигнала приводит не только к битовым ошибкам передаваемой информации, но и к *неопределенности зоны покрытия* беспроводной локальной сети. В проводных локальных сетях такой проблемы нет, те и только те устройства, которые подключены к кабельной системе здания или кампуса, получают сигналы и участвуют в работе LAN. Беспроводная локальная сеть не имеет точной области покрытия. Часто используемое изображение такой области в форме шестиугольника или круга является не чем иным, как абстракцией. В действительности, сигнал может быть настолько ослаблен, что устройства, находящиеся в предполагаемых пределах зоны покрытия, вообще не могут принимать и передавать информацию.

Рисунок 12.13 хорошо иллюстрирует такую ситуацию. Подчеркнем, что с течением времени ситуация с распределением сигнала может измениться вместе с изменением состава LAN. По этой причине даже технологии, рассчитанные на фиксированные (не мобильные) узлы сети, должны учитывать то, что беспроводная локальная сеть является неполносвязной. Даже если считать, что сигнал распространяется идеально во все стороны, образованию полносвязной топологии может мешать то, что радиосигнал затухает пропорционально квадрату расстояния от источника. Поэтому при отсутствии базовой станции некоторые пары узлов не смогут взаимодействовать из-за того, что расположены за пределами зоны покрытия передатчиков партнера.

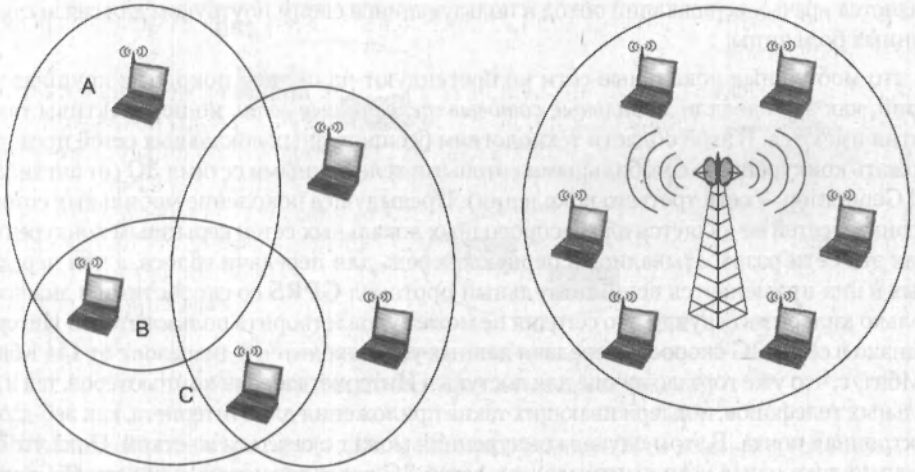
В примере на рис. 12.14, *а* показана такая фрагментированная локальная сеть. Неполносвязность беспроводной сети порождает проблему доступа к разделяемой среде, известную под названием **скрытого терминала**. Проблема возникает в том случае, когда два узла находятся вне зон досягаемости друг друга (узлы *A* и *C* на рис. 12.14, *а*), но существует третий узел *B*, который принимает сигналы как от *A*, так и от *C*. Предположим, что в радиосети используется традиционный метод доступа, основанный на прослушивании несущей, например CSMA/CD. В данном случае коллизии будут возникать значительно чаще, чем в проводных сетях. Пусть, например, узел *B* занят обменом с узлом *A*. Узлу *C* сложно определить, что среда занята, он может посчитать ее свободной и начать передавать свой кадр. В результате сигналы в районе узла *B* исказятся, то есть произойдет коллизия, вероятность возникновения которой в проводной сети была бы неизмеримо ниже.

Распознавание коллизий затруднено в радиосети еще и потому, что сигнал собственного передатчика существенно подавляет сигнал удаленного передатчика, и распознать искажение сигнала чаще всего невозможно.

> В методах доступа, применяемых в беспроводных сетях, отказываются не только от прослушивания несущей, но и от распознавания коллизий.

Вместо этого в них используют методы предотвращения коллизий, включая **методы опроса**.

Применение базовой станции может улучшить связность сети (рис. 12.14, *б*). Базовая станция обычно обладает большей мощностью, а ее антенна устанавливается так, чтобы более равномерно и беспрепятственно покрывать нужную территорию. В результате все узлы беспроводной локальной сети получают возможность обмениваться данными с базовой станцией, которая транзитом передает данные между узлами.



а *б*

Рис. 12.14. Связность беспроводной локальной сети: *а* — специализированная беспроводная сеть, *б* — беспроводная сеть с базовой станцией

Беспроводные локальные сети считаются перспективными для таких применений, в которых сложно или невозможно использовать проводные сети. Далее перечислены основные области применения беспроводных локальных сетей.

- *Домашние локальные сети.* Когда в доме появляется несколько компьютеров, организация домашней локальной сети становится насущной проблемой. Пользователи быстро поняли преимущества беспроводных домашних сетей, не требующих прокладки в квартире или доме кабеля на витой паре и позволяющих легко переносить ноутбук из комнаты в комнату. Производители также быстро отреагировали на этот спрос, приступив к выпуску для таких сетей компактных многофункциональных центральных устройств, совмещающих функции модема, маршрутизатора и точки беспроводного доступа. Практически все современные ноутбуки имеют сегодня встроенные беспроводные сетевые адаптеры, ими также оснащены многие принтеры.
- *Резидентный доступ альтернативных операторов связи*, у которых нет проводного доступа к клиентам, проживающим в многоквартирных домах.
- Так называемый *«кочевой» доступ* в аэропортах, железнодорожных вокзалах и т. п.
- Организация локальных сетей в зданиях, где нет возможности установить современную кабельную систему, например *в исторических зданиях* с оригинальным интерьером.
- Организация *временных локальных сетей*, например, при проведении конференций.
- *Расширения локальных сетей.* Иногда одно здание предприятия, например испытательная лаборатория или цех, может быть расположено отдельно от других. Небольшое число рабочих мест в таком здании делает крайне невыгодным прокладку к нему отдельного кабеля, поэтому беспроводная связь оказывается более рациональным вариантом.
- *Мобильные локальные сети.* Если пользователь хочет получать услуги сети, перемещаясь из помещения в помещение или из здания в здание, то здесь конкурентов у беспроводной локальной сети просто нет. Классическим примером такого пользователя

является врач, совершающий обход и пользующийся своим ноутбуком для связи с базой данных больницы.

Пока что мобильные локальные сети не претендуют на полное покрытие крупных территорий, как это сделали *мобильные сотовые телефонные сети*, но перспективы такого развития имеются. В этой области технологиям беспроводных локальных сетей предстоит выдержать конкуренцию с мобильными сотовыми телефонными сетями **3G** (от английского 3rd Generation — сети третьего поколения). Предыдущее поколение мобильных сотовых телефонных сетей не является для беспроводных локальных сетей серьезным конкурентом, так как эти сети разрабатывались в первую очередь для передачи голоса, а для передачи данных в них применяется вспомогательный протокол **GPRS** со скоростями в диапазоне несколько килобит в секунду, что сегодня не может удовлетворить пользователей Интернета. Однако в сетях 3G скорость передачи данных уже находится в диапазоне от 144 Кбит/с до 2 Мбит/с, что уже гораздо лучше для доступа в Интернет как для компьютеров, так и для мобильных телефонов, поддерживающих такие приложения для Интернета, как веб-доступ и электронная почта. В этом случае конкуренция может оказаться жесткой. Пока что беспроводные локальные сети выигрывают у сетей 3G соревнование в скорости (54 против 2 Мбит/с), но уступают в мобильности, так как их область покрытия обычно ограничена зданием или небольшой территорией аэропорта или вокзала.

Далее будет рассмотрен самый популярный стандарт беспроводных локальных сетей — **IEEE 802.11**. Сети и оборудование IEEE.802.11 также известны под названием **Wi-Fi** — по имени консорциума Wi-Fi[1] Alliance (http://wi-fi.org), который занимается вопросам совместимости и сертификации оборудования стандартов IEEE 802.11.

Топологии локальных сетей стандарта 802.11

Стандарт 802.11 поддерживает два типа топологий локальных сетей: с базовым и с расширенным наборами услуг.

Сеть с **базовым набором услуг** (Basic Service Set, BSS) образуется отдельными станциями, базовая станция отсутствует, узлы взаимодействуют друг с другом непосредственно (рис. 12.15). Для того чтобы войти в сеть BSS, станция должна выполнить процедуру присоединения.

Сети BSS не являются традиционными сотами в отношении зон покрытия, они могут находиться друг от друга на значительном расстоянии, а могут частично или полностью перекрываться — стандарт 802.11 оставляет здесь свободу для проектировщика сети.

Станции могут использовать разделяемую среду для того, чтобы передавать данные:

- непосредственно друг другу в пределах одной сети BSS;
- в пределах одной сети BSS транзитом через точку доступа;
- между разными сетями BSS через две точки доступа и распределенную систему;
- между сетью BSS и проводной локальной сетью через точку доступа, распределенную систему и портал[2].

---

[1] Wi-Fi является сокращением от Wireless Fidelity — «беспроводная точность»; термин был введен по аналогии с популярным термином Hi-Fi, обозначающим высокую точность воспроизведения звука аппаратурой.

[2] Функции портала стандартом не детализируются, это может быть коммутатор или маршрутизатор.

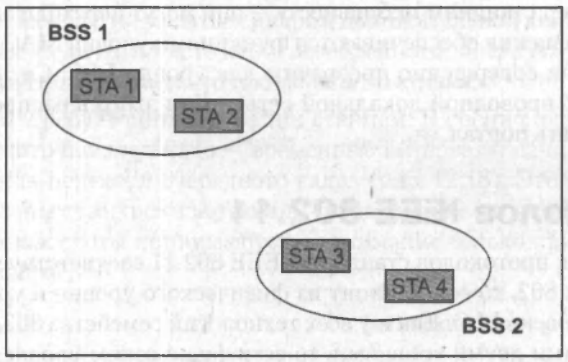


Рис. 12.15. Сети с базовым набором услуг

В сетях, обладающих инфраструктурой, некоторые станции сети являются базовыми, или, в терминологии 802.11, **точками доступа** (Access Point, AP). Станция, которая выполняет функции AP, является членом какой-нибудь сети BSS (рис. 12.16). Все базовые станции сети связаны между собой с помощью распределенной системы (Distribution System, DS), в качестве которой может использоваться та же среда (то есть радио- или инфракрасные волны), что и среда взаимодействия между станциями, или же отличная от нее, например проводная. Точки доступа вместе с распределенной системой поддерживают **службу распределенной системы** (Distribution System Service, DSS). Задачей DSS является передача пакетов между станциями, которые по каким-то причинам не могут или не хотят взаимодействовать между собой непосредственно. Наиболее очевидной причиной использования DSS является принадлежность станций разным сетям BSS. В этом случае они передают кадр своей точке доступа, которая через DS передает его точке доступа, обслуживающей сеть BSS со станцией назначения.

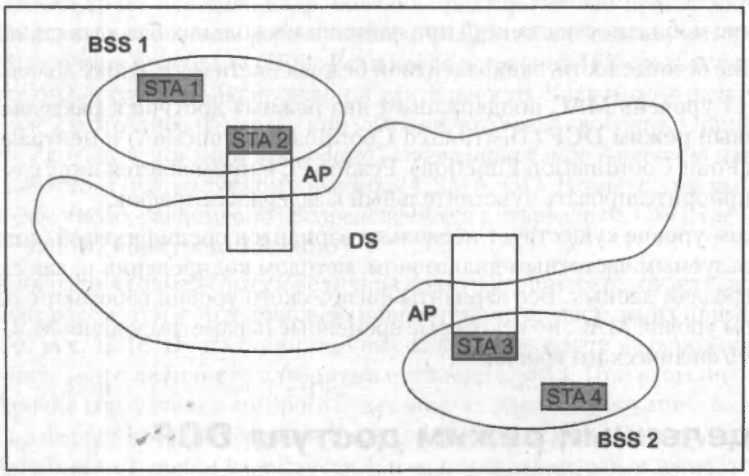


Рис. 12.16. Сеть с расширенным набором услуг

Сеть с **расширенным набором услуг** (Extended Service Set, ESS) состоит из нескольких сетей BSS, объединенных распределенной средой.

Сеть ESS обеспечивает станциям мобильность — они могут переходить из одной сети BSS в другую. Эти перемещения обеспечиваются функциями уровня MAC рабочих и базовых станций, поэтому они совершенно прозрачны для уровня LLC. Сеть ESS может также взаимодействовать с проводной локальной сетью. Для этого в распределенной системе должен присутствовать портал.

Стек протоколов IEEE 802.11

Естественно, что стек протоколов стандарта IEEE 802.11 соответствует общей структуре стандартов комитета 802, то есть состоит из физического уровня и уровня MAC, поверх которых работает уровень LLC. Как и у всех технологий семейства 802, технология 802.11 определяется нижними двумя уровнями, то есть физическим уровнем и уровнем MAC, а уровень LLC выполняет свои стандартные общие для всех технологий LAN функции.

Структура стека протоколов IEEE 802.11 показана на рис. 12.17.

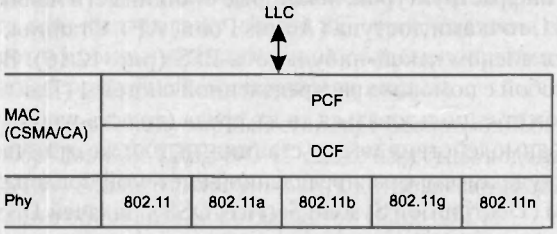


Рис. 12.17. Стек протоколов IEEE 802.11

Уровень MAC выполняет в беспроводных сетях больше функций, чем в проводных. Функции уровня MAC в стандарте 802.11 включают:

- доступ к разделяемой среде;
- обеспечение мобильности станций при наличии нескольких базовых станций;
- обеспечение безопасности, эквивалентной безопасности проводных локальных сетей.

В сетях 802.11 уровень MAC поддерживает два режима доступа к разделяемой среде: **распределенный режим DCF** (Distributed Coordination Function) и **централизованный режим PCF** (Point Coordination Function). Режим PCF применяется в тех случаях, когда необходимо приоритезировать чувствительный к задержкам трафик.

На физическом уровне существует несколько вариантов спецификаций, которые отличаются используемым частотным диапазоном, методом кодирования и, как следствие, – скоростью передачи данных. Все варианты физического уровня работают с одним и тем же алгоритмом уровня MAC, но некоторые временные параметры уровня MAC зависят от используемого физического уровня.

Распределенный режим доступа DCF

Рассмотрим сначала, как обеспечивается доступ в распределенном режиме DCF. В этом режиме реализуется метод **CSMA/CA** (Carrier Sense Multiple Access with Collision Avoidance — метод прослушивания несущей частоты с множественным доступом и предотвращением коллизий). Вместо неэффективного в беспроводных сетях прямого распознавания коллизий по методу CSMA/CD здесь они выявляются косвенно. Для этого каждый

переданный кадр должен подтверждаться **кадром положительной квитанции**, посылаемым станцией назначения. Если же по истечении оговоренного тайм-аута квитанция не поступает, станция-отправитель считает, что произошла коллизия.

Режим доступа DCF требует синхронизации станций. В спецификации 802.11 эта проблема решается достаточно элегантно — временные интервалы начинают отсчитываться от момента окончания передачи очередного кадра (рис. 12.18). Это не требует передачи каких-либо специальных синхронизирующих сигналов и не ограничивает размер пакета размером слота, так как слоты принимаются во внимание только при принятии решения о начале передачи кадра.

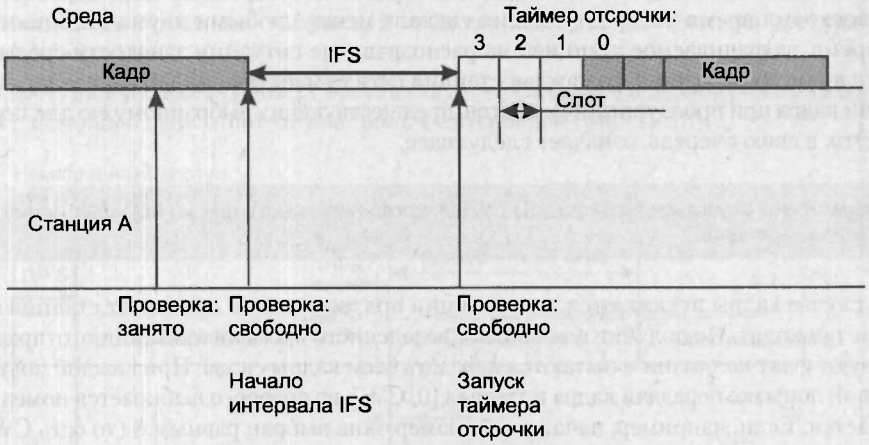


Рис. 12.18. Режим доступа DCF

Станция, которая хочет передать кадр, обязана предварительно прослушать среду. Как только она фиксирует окончание передачи кадра, она обязана отсчитать интервал времени, равный межкадровому интервалу (IFS). Если после истечения IFS среда все еще свободна, то начинается отсчет слотов фиксированной длительности. Кадр можно начать передавать только в начале какого-либо из слотов при условии, что среда свободна. Станция выбирает для передачи слот на основании *усеченного экспоненциального двоичного алгоритма отсрочки*, аналогичного используемому в методе CSMA/CD. Номер слота выбирается как случайное целое число, равномерно распределенное в интервале [0, CW], где CW означает Contention Window (**конкурентное окно**).

О том, как выбирается размер слота и величина конкурентного окна, будет сказано немного позже, а сейчас рассмотрим этот довольно непростой метод доступа на примере, который иллюстрирует рис. 12.18. Пусть станция *A* выбрала для передачи на основании усеченного экспоненциального двоичного алгоритма отсрочки слот 3. При этом она присваивает **таймеру отсрочки** (назначение которого будет ясно из дальнейшего описания) значение 3 и начинает проверять состояние среды в начале каждого слота. Если среда свободна, то из значения таймера отсрочки вычитается 1, и если результат равен нулю, то начинается передача кадра.

Таким образом обеспечивается условие незанятости всех слотов, включая выбранный. Это условие является необходимым для начала передачи.

Если же в начале какого-нибудь слота среда оказывается занятой, то вычитания единицы не происходит, и таймер «замораживается». В этом случае станция начинает новый цикл доступа к среде, изменяя только алгоритм выбора слота для передачи. Как и в предыдущем цикле, станция следит за средой и при ее освобождении делает паузу в течение межкадрового интервала. Если среда осталась свободной, то станция *использует значение «замороженного» таймера в качестве номера слота* и выполняет описанную процедуру проверки свободных слотов с вычитанием единиц, начиная с замороженного значения таймера отсрочки.

Размер слота зависит от способа кодирования сигнала; так, для метода FHSS размер слота равен 28 мкс, а для метода DSSS — 1 мкс. Размер слота выбирается таким образом, чтобы он превосходил время распространения сигнала между любыми двумя станциями сети плюс время, затрачиваемое станцией на распознавание ситуации занятости среды. Если такое условие соблюдается, то каждая станция сети сумеет правильно распознать начало передачи кадра при прослушивании слотов, предшествующих выбранному ею для передачи слоту. Это, в свою очередь, означает следующее.

Коллизия может случиться только в том случае, когда несколько станций выбирают один и тот же слот для передачи.

В этом случае кадры искажаются, и квитанции подтверждения приема от станций назначения не приходят. Не получив в течение определенного времени квитанцию, отправители фиксируют факт коллизии и пытаются передать свои кадры снова. При каждой повторной неудачной попытке передачи кадра интервал [0, CW], из которого выбирается номер слота, удваивается. Если, например, начальный размер окна выбран равным 8 (то есть CW = 7), то после первой коллизии размер окна должен быть равен 16 (CW = 15), после второй последовательной коллизии — 32 и т. д. Начальное значение CW в соответствии со стандартом 802.11 должно выбираться в зависимости от типа физического уровня, используемого в беспроводной локальной сети.

Как и в методе CSMA/CD, в данном методе количество неудачных попыток передачи одного кадра ограничено, но стандарт 802.11 не дает точного значения этого верхнего предела. Когда верхний предел в N попыток достигнут, то кадр отбрасывается, а счетчик последовательных коллизий устанавливается в нуль. Этот счетчик также устанавливается в нуль, если кадр после некоторого количества неудачных попыток все же передается успешно.

ПРИМЕЧАНИЕ

Максимальная длина кадра данных 802.11 равна 2346 байт, длина кадра RTS — 20 байт, кадра CTS — 14 байт. Так как кадры RTS и CTS гораздо короче, чем кадр данных, то потери данных в результате коллизии кадров RTS или CTS гораздо меньше, чем при коллизии кадров данных. Процедура обмена кадрами RTS и CTS не обязательна. От нее можно отказаться при небольшой нагрузке сети, поскольку в такой ситуации коллизии случаются редко, а значит, не стоит тратить дополнительное время на выполнение процедуры обмена кадрами RTS и CTS.

В режиме доступа DFC применяются меры для *устранения эффекта скрытого терминала*. Для этого станция, которая хочет захватить среду и в соответствии с описанным алгоритмом начинает передачу кадра в определенном слоте, вместо кадра данных сначала посылает станции назначения короткий служебный кадр RTS (Request To Send — запрос

на передачу). На этот запрос станция назначения должна ответить служебным кадром CTS (Clear To Send – свободна для передачи), после чего станция-отправитель посылает кадр данных. Кадр CTS должен оповестить о захвате среды те станции, которые находятся вне зоны сигнала станции-отправителя, но в зоне досягаемости станции-получателя, то есть являются скрытыми терминалами для станции-отправителя.

Централизованный режим доступа PCF

В том случае, когда в сети BSS имеется станция, выполняющая функции точки доступа, может применяться также централизованный режим доступа PCF, обеспечивающий приоритетное обслуживание трафика. В этом случае говорят, что точка доступа играет роль арбитра среды.

Режим доступа PCF в сетях 802.11 сосуществует с режимом DCF. Оба режима координируются с помощью трех типов межкадровых интервалов (рис. 12.19).

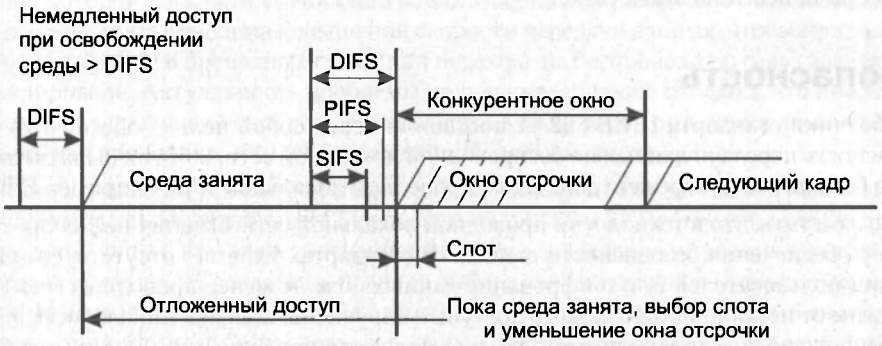


Рис. 12.19. Сосуществование режимов PCF и DCF

После освобождения среды каждая станция отсчитывает время простоя среды, сравнивая его с тремя значениями:

- короткий межкадровый интервал (Short IFS, **SIFS**);
- межкадровый интервал режима PCF (**PIFS**);
- межкадровый интервал режима DCF (**DIFS**).

Захват среды с помощью распределенной процедуры DCF возможен только в том случае, когда среда свободна в течение времени, равного или большего, чем DIFS. То есть в качестве IFS в режиме DCF нужно использовать интервал DIFS – самый длительный период из трех возможных, что дает этому режиму самый низкий приоритет.

Межкадровый интервал SIFS имеет наименьшее значение, он служит для первоочередного захвата среды ответными кадрами CTS или квитанциями, которые продолжают или завершают уже начавшуюся передачу кадра.

Значение межкадрового интервала PIFS больше, чем SIFS, но меньше, чем DIFS. Промежутком времени между завершением PIFS и DIFS пользуется арбитр среды. В этом промежутке он может передать специальный кадр, который говорит всем станциям, что начинается **контролируемый период**. Получив этот кадр, станции, которые хотели бы воспользоваться алгоритмом DCF для захвата среды, уже не могут этого сделать, они должны

дожидаться окончания контролируемого периода. Длительность этого периода объявляется в специальном кадре, но этот период может закончиться и раньше, если у станций нет чувствительного к задержкам трафика. В этом случае арбитр передает служебный кадр, после которого по истечении интервала DIFS начинает работать режим DCF.

На управляемом интервале реализуется *централизованный метод* доступа PCF. Арбитр выполняет процедуру опроса, чтобы по очереди предоставить каждой такой станции право на использование среды, направляя ей специальный кадр. Станция, получив такой кадр, может ответить другим кадром, который подтверждает прием специального кадра и одновременно передает данные (либо по адресу арбитра для транзитной передачи, либо непосредственно станции).

Для того чтобы какая-то доля среды всегда доставалась асинхронному трафику, длительность контролируемого периода ограничена. После его окончания арбитр передает соответствующий кадр, и начинается неконтролируемый период.

Каждая станция может работать в режиме PCF, для этого она должна подписаться на эту услугу при присоединении к сети.

Безопасность

Разработчики стандарта IEEE 802.11 поставили перед собой цель — обеспечить такую безопасность передачи данных по беспроводной локальной сети, которая была бы эквивалентна безопасности передачи данных по проводной локальной сети, например Ethernet.

Можно заметить, что в технологии проводной локальной сети Ethernet нет каких-то особых мер обеспечения безопасности данных. В стандартах Ethernet отсутствует аутентификация пользователей или шифрование данных. Тем не менее проводные сети лучше защищены от несанкционированного доступа и нарушения конфиденциальности данных, чем беспроводные — только потому, что они являются проводными. Действительно, для того чтобы получить доступ к проводной сети, злоумышленник должен к ней физически присоединиться. Для этого ему нужно каким-то образом попасть в помещение, где имеются розетки, и присоединить свой компьютер к одной из них. Такое действие можно заметить и пресечь (хотя возможности для несанкционированного доступа к проводной сети все равно остаются).

В беспроводной сети несанкционированный доступ можно осуществить гораздо проще, достаточно оказаться в зоне распространения радиоволн этой сети. Для этого можно даже не входить в здание, где развернута сеть. Физическое подключение к среде в этом случае также не требуется, так что посетитель может принимать данные, не производя подозрительных действий, а просто имея работающий ноутбук в своей сумке.

В стандарте 802.11 предусмотрены средства обеспечения безопасности, которые повышают защищенность беспроводной локальной сети до уровня обычной проводной локальной сети. Поэтому основной протокол защиты данных в сетях 802.11 так и называется — **WEP** (Wired Equivalent Privacy — секретность, эквивалентная проводной). Он предоставляет возможность шифровать данные, передаваемые через беспроводную среду, и тем самым обеспечивает их конфиденциальность. Технология 802.11 предлагает еще один механизм безопасности — аутентификацию — доказательство легальности пользователя, подключающегося к сети. Однако несовершенство средств безопасности 802.11 делают их популярной мишенью для критиков. Например, исследуя зашифрованный трафик 802.11, взломщик может расшифровать информацию в течение 24 часов.

Для разработки более защищенного варианта беспроводных локальных сетей была создана рабочая группа 802.11i. В 2003 году консорциум Wi-Fi Alliance выпустил спецификацию под названием WPA (Wi-Fi Protected Access — защищенный доступ к Wi-Fi), которая представляла собой промежуточный неокончательный вариант стандарта 802.11i. В результате окончательный вариант стандарта 802.11i, одобренный в 2004 году, получил неофициальное название WPA2. Стандарт WPA2 описывает надежное средство защиты беспроводных локальных сетей, сочетающее в себе наиболее совершенные средства аутентификации пользователей и шифрования данных, применимые в компьютерных сетях. Поддержка протокола WPA2 является необходимым условием сертификации оборудования консорциумом Wi-Fi Alliance.

Физические уровни стандарта 802.11

С момента принятия первой версии стандарта 802.11 в 1997 году одной из главных проблем, над которой работали специалисты, занимающиеся развитием беспроводных локальных сетей, была проблема повышения скорости передачи данных, чтобы приложения, хорошо работающие в проводных сетях, при переходе на беспроводную связь значительно не деградировали. Актуальность проблемы подчеркивает также тот факт, что пропускная способность беспроводной сети всегда разделяется между всеми пользователями этой сети, в то время как проводные сети уже ушли от разделяемой среды.

Другой немаловажной проблемой является выбранный диапазон частот радиоспектра. В соответствии с рекомендациями ITU диапазоны 2,4, 3,6 и 5 ГГц отведены для беспроводной передачи данных, при этом лицензирование этих диапазонов не рекомендуется. В разных странах существуют различные правила выбора этих диапазонов (причем правила для каждого из диапазонов могут быть разными), от свободного использования до обычного лицензирования. Помимо беспроводных локальных сетей в этих диапазонах могут работать и другие типы устройств, например любительское радио или беспроводные сети городов. В США диапазон 3,6 ГГц сравнительно недавно был отведен для беспроводных локальных сетей, в то время как в Европе он уже в течение ряда лет выделен для беспроводных сетей городов, работающих по стандарту IEEE 802.16[1] (WiMAX).

Физические уровни стандарта 802.11 1997 года

В 1997 году комитетом **802.11** был принят стандарт, который определял функции уровня MAC вместе с *тремя вариантами физического уровня*, которые обеспечивают передачу данных со скоростями 1 и 2 Мбит/с.

- В первом варианте средой являются *инфракрасные волны* диапазона 850 нм, которые генерируются либо полупроводниковым лазерным диодом, либо светодиодом (LED). Так как инфракрасные волны не проникают через стены, область покрытия LAN ограничивается зоной прямой видимости. Стандарт предусматривает три варианта распространения излучения: ненаправленную антенну, отражение от потолка и фокусное направленное излучение. В первом случае узкий луч рассеивается с помощью системы линз. Фокусное направленное излучение предназначено для организации двухточечной связи, например между двумя зданиями.

[1] Этот стандарт будет кратко рассмотрен в главе 22.

- Во втором варианте в качестве передающей среды используется *микроволновый диапазон* 2,4 ГГц. Этот вариант основан на методе FHSS (см. главу 10). В методе FHSS каждый узкий канал имеет ширину 1 МГц. Частотная манипуляция (FSK) с двумя состояниями сигнала (частотами) дает скорость 1 Мбит/с, с четырьмя состояниями – 2 Мбит/с. В случае FHSS сеть может состоять из сот, причем для исключения взаимного влияния в соседних сотах могут применяться ортогональные последовательности частот. Количество каналов и частота переключения между каналами настраиваются, так что при развертывании беспроводной локальной сети можно учитывать особенности регулирования спектра частот конкретной страны.
- Третий вариант, в котором используется тот же *микроволновый диапазон*, основан на методе DSSS, где в качестве последовательности чипов применяется 11-битный код 10110111000. Каждый бит кодируется путем двоичной фазовой (1 Мбит/с) или квадратурной фазовой (2 Мбит/с) манипуляции.

Физические уровни стандартов 802.11a и 802.11b

В 1999 году были приняты два варианта стандарта физического уровня: **802.11a** и **802.11b**, заменяющие спецификации физического уровня 802.11 редакции 1997.

В *спецификации 802.11b института IEEE* по-прежнему используется диапазон 2,4 ГГц. Для повышения скорости до 11 Мбит/с, которая сопоставима со скоростью классического стандарта Ethernet, здесь применяется более эффективный вариант метода DSSS, опирающийся на технику Complementary Code Keying (CCK), заменившую коды Баркера.

Однако диапазон 2,4 ГГц с шириной полосы примерно в 80 МГц используется стандартом 802.11b отличным от стандарта 1997 года способом. Этот диапазон разбит на 14 каналов, каждый из которых, кроме последнего, отстоит от соседей на 5 МГц (рис. 12.20).

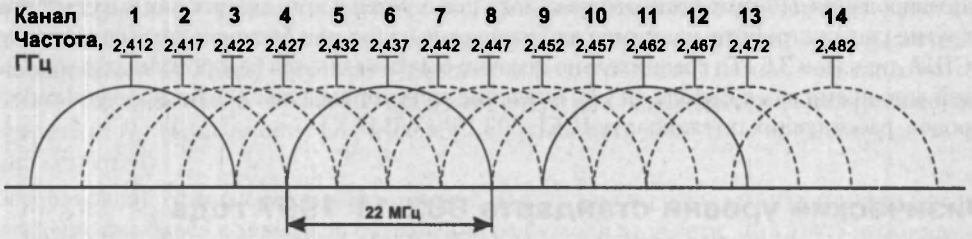


Рис. 12.20. Разбиение диапазона 2,4 ГГц на каналы

Для передачи данных согласно стандарту 802.11b используется полоса частот шириной в 22 МГц, поэтому одного канала шириной в 5 МГц оказывается недостаточно, приходится объединять несколько соседних каналов. Для того чтобы гарантировать некоторый минимум взаимных помех, возникающих от передатчиков, работающих в диапазоне 2,4 ГГц, комитет 802.11 определил так называемую спектральную маску, определяющую разрешенный спектр мощности передатчика, работающего в каком-либо из каналов. Это спектр должен затухать не меньше чем на 30 дБ на расстоянии 11 МГц от центра канала, что и создает укрупненную полосу шириной в 22 МГц с центром в некотором из 14 каналов.

В результате одновременно в одной и той же области покрытия могут работать несколько независимых беспроводных сетей стандарта 802.11b. На рис. 12.20 показан вариант для трех сетей, использующих каналы 1, 6 и 11. Такое использование каналов типично для

США, где частотные каналы 12, 13 и 14 для сетей стандарта 802.11 не разрешены. В Европе в конце 90-х годов действовали более жесткие ограничения, например, в Испании были разрешены только каналы 10 и 11, а во Франции — только каналы 10, 11, 12 и 13, но постепенно эти ограничения были сняты, и сейчас лишь канал 14 в большинстве стран по-прежнему не задействован. Таким образом, в странах Европы максимальное количество независимых сетей, работающих в одной области покрытия, достигает 4; обычно они используют каналы 1, 5, 9 и 13.

Оборудование стандарта 802.11b может конфигурироваться для любого из 14 каналов диапазона 2,4 ГГц, так что при возникновении помех на определенном канале можно перейти на другой.

Спецификация 802.11a обеспечивает повышение скорости передачи данных за счет использования полосы частот шириной 300 МГц из диапазона частот 5 ГГц. Так как полоса частот, отведенная для беспроводных локальных сетей, в этом диапазоне шире, то и количество каналов шириной в 5 МГц здесь больше, чем в диапазоне 2,4 ГГц — в зависимости от правил регулирования конкретной страны их может быть 48 и более. Для передачи данных в технологии задействована полоса частот шириной 20 МГц, что дает возможность иметь 12 и более независимых сетей в одной области покрытия.

Для кодирования данных в стандарте 802.11a используется техника ортогонального частотного мультиплексирования (OFDM). Данные первоначально кодируются на 52 первичных несущих частотах методом BPSK, QPSK, 16-QAM или 64-QAM, а затем сворачиваются в общий сигнал с шириной спектра в 20 МГц. Скорость передачи данных в зависимости от метода кодирования первичной несущей частоты составляет 6, 9, 12, 18, 24, 36, 48 или 54 Мбит/с.

Диапазон 5 ГГц в спецификации 802.11a пока меньше «населен» и предоставляет больше частотных каналов для передачи данных. Однако его использование связано с несколькими проблемами. Во-первых, оборудование для этих частот пока еще слишком дорогое, во-вторых, в некоторых странах частоты этого диапазона подлежат лицензированию, в-третьих, волны этого диапазона хуже проходят через препятствия.

Физический уровень стандарта 802.11g

Стандарт **802.11g** для физического уровня разработан рабочей группой института IEEE летом 2003 года. Он быстро завоевал популярность, так как обеспечивал те же скорости, что и стандарт 802.1a, то есть до 54 Мбит/с, но в диапазоне 2,4 ГГц, то есть в том диапазоне, где до этого удавалось достигать максимальной скорости в 11 Мбит/с на оборудовании стандарта 802.11b. В то же время стоимость оборудования стандарта 802.1g достаточно быстро стала соизмеримой со стоимостью оборудования стандарта 802.11b, что и стало причиной роста популярности новой спецификации. В ней, так же как и в спецификации 802.11a, используется ортогональное частотное мультиплексирование (OFDM). До некоторого времени в США в диапазоне 2,4 ГГц разрешалось применять только технику расширения спектра, такую как FSSS или DSSS. Снятие этого ограничения дало импульс разработкам, в результате появилась новая высокоскоростная беспроводная технология для этого диапазона частот. Для обратной совместимости со стандартом 802.11b поддерживается также техника CCK.

Диаметр сети стандарта 802.11 зависит от многих параметров, в том числе от используемого диапазона частот. Обычно диаметр беспроводной локальной сети находится в пределах от 100 до 300 м вне помещений и от 30 до 40 м внутри помещений.

В 2007 году стандарты 802.11a и 802.11b были сведены в новую редакцию стандарта 802.11-2007, где каждому из них отведен соответствующий раздел.

Физический уровень стандарта 802.11n

Стандарт 802.1n, работы над которым были начаты еще в 2004 году, на момент написания этой книги еще не был окончательно утвержден, хотя такое событие ожидалось уже к концу 2008 года, затем было отложено до конца 2009 года, а теперь согласно последним сведениям перенесено на начало 2010.

Тем не менее оборудование «pre-N» в соответствии с версией 2 проекта стандарта 802.11n появилось на рынке в конце 2006 года, а с начала 2007 года консорциум Wi-Fi Alliance начал его сертификацию.

Основной особенностью стандарта 802.11n является дальнейшее повышение скорости передачи данных (до 300 Мбит/с и выше). Оборудование стандарта 802.11n может работать как в диапазоне 5 ГГц, так и в диапазоне 2,4 ГГц, хотя рекомендуемым диапазоном является диапазон 5 ГГц благодаря большему числу доступных каналов и меньшей интерференции с многочисленным оборудованием, работающим сегодня в диапазоне 2,4 ГГц.

Для достижения высоких скоростей в технологии 802.11n применено несколько новых механизмов.

- *Улучшенное кодирование OFDM и сдвоенные частотные каналы.* Вместо каналов с полосой в 20 МГц, которые использовались в технологиях 802.11a и 802.11g, в технологии 802.11n применены каналы с полосой 40 МГц (для обратной совместимости допускается также работать с каналами 20 МГц). Само по себе расширение полосы в два раза должно приводить к повышению битовой скорости в два раза, но выигрыш здесь больше за счет усовершенствований в кодировании OFDM: вместо 52 первичных несущих частот на полосу в 20 МГц здесь используется 57 таких частот, а на полосу в 40 МГц соответственно 114. Это приводит к повышению битовой скорости с 54 до 65 Мбит/с для каналов 20 МГц и до 135 Мбит/с для каналов 40 МГц.
- *Уменьшение межсимвольного интервала.* Для надежного распознавания кодовых символов в технологиях 80.11a/g используется межсимвольный интервал в 800 нс. Технология 802.11n позволяет передавать данные с таким же межсимвольным интервалом, а также с межсимвольным интервалом в 400 нс, что повышает битовую скорость для каналов 40 МГц до 150 Мбит/с.
- *Применение техники MIMO* (Multiple Input Multiple Output — множественные входы и выходы). Эта техника основана на использовании одним сетевым адаптером нескольких антенн с целью лучшего распознавания сигнала, пришедшего к приемнику разными путями. Обычно из-за таких эффектов распространения радиоволн, как отражение, дифракция и рассеивание, приемник получает несколько сигналов, дошедших от передатчика по разным физическим путям и имеющим, следовательно, сдвиг по фазе. До введения техники MIMO такие явления считались негативными и с ними боролись путем применения нескольких (обычно двух) антенн, из которых в каждый момент времени использовалась только одна — та, которая принимала сигнал лучшего качества. Техника MIMO принципиально изменила отношение к сигналам, пришедшим разными путями, — эти сигналы комбинируются и путем цифровой обработки из них восстанавливается исходный сигнал.

Техника MIMO не только способствует улучшению соотношения сигнал/помеха. Благодаря возможности обрабатывать сигналы, пришедшие разными путями, для создания избыточного сигнала для каждого потока можно передавать с помощью нескольких антенн несколько независимых потоков данных (обычно их число меньше, чем число антенн). Эта способность систем MIMO называется **пространственным мультиплексированием** (spatial multiplexing). Для систем MIMO принято использовать обозначение:

$$T \times R : S.$$

Здесь T — количество передающих антенн узла, R — количество принимающих антенн узла, а S — количество потоков данных, которые пространственно мультиплексируются. Типичной системой MIMO в выпускаемом в 2009 году оборудовании стандарта 802.11n является система $3 \times 3 : 2$, то есть система с тремя передающими и тремя принимающими антеннами, которая позволяет передавать два независимых потока данных. Система MIMO $3 \times 3 : 2$ обеспечивает повышение битовой скорости в два раза, то есть до 300 Мбит/с для каналов 40 МГц.

Проект стандарта 802.11 предусматривает различные варианты системы MIMO вплоть до $4 \times 4 : 4$, что позволило бы повысить битовую скорость до 600 Мбит/с.

Помимо усовершенствований физического уровня, стандарт 802.11n вводит одно усовершенствование на уровне MAC — это возможность агрегирования нескольких кадров данных в один кадр. Такая техника повышает эффективность передачи пользовательских данных при той же битовой скорости протокола за счет сокращения накладных расходов на шифрование отдельных кадров и на их индивидуальное подтверждение положительными квитанциями со случайными паузами между передачей кадров. Кроме того, для мультимедийных приложений допускается уменьшение интервала DIFS при передаче длительной пульсации трафика.

Персональные сети и технология Bluetooth

Особенности персональных сетей

Персональные сети (Personal Area Network, PAN) предназначены для взаимодействия устройств, принадлежащих одному владельцу, на небольшом расстоянии, обычно *в радиусе 10 м*. Такими устройствами могут быть ноутбук, мобильный телефон, принтер, карманный компьютер (Personal Digital Assistant, PDA), телевизор, а также многочисленные бытовые приборы, например холодильник.

Персональные сети предназначены для соединения устройств, принадлежащих, как правило, одному пользователю, на небольших расстояниях. Типичным примером PAN является беспроводное соединение компьютера с периферийными устройствами, такими как принтер, наушники, мышь, клавиатура и т. п. Мобильные телефоны также используют технологию PAN для соединения со своей периферией (чаще всего это наушники), а также с компьютером своего владельца. Некоторые марки наручных часов стали поддерживать технологию PAN, превращаясь в универсальные устройства с функциями PDA.

Персональные сети должны обеспечивать как фиксированный доступ, например, в пределах дома, так и мобильный, когда владелец устройств PAN перемещается вместе с ними между помещениями или городами.

Персональные сети во многом похожи на локальные, но у них есть и свои особенности.

- Многие из устройств, которые могут входить в персональную сеть, *гораздо проще*, чем традиционный узел LAN — компьютер. Кроме того, такие устройства обычно имеют небольшие габариты и стоимость. Поэтому стандарты PAN должны учитывать, что их реализация должна приводить к недорогим решениям, потребляющим небольшую энергию.
- *Область покрытия PAN меньше области покрытия LAN*, узлы PAN часто находятся на расстоянии нескольких метров друг от друга.
- *Высокие требования к безопасности.* Персональные устройства, путешествуя вместе со своим владельцем, попадают в различное окружение. Иногда они должны взаимодействовать с устройствами других персональных сетей, например, если их владелец встретил на улице своего знакомого и решил переписать из его устройства PDA в свое несколько адресов общих знакомых. В других случаях такое взаимодействие явно нежелательно, так как может привести к утечке конфиденциальной информации. Поэтому протоколы PAN должны обеспечивать разнообразные методы аутентификации устройств и шифрования данных в мобильной обстановке.
- При соединении малогабаритных устройств между собой желание избавиться от кабелей проявляется гораздо сильнее, чем при соединении компьютера с принтером или концентратором. Из-за этого персональные сети в гораздо большей степени, чем локальные, *тяготеют к беспроводным решениям*.
- Если человек постоянно носит устройство PAN с собой и на себе, то оно не должно причинять вред его здоровью. Поэтому такое устройство должно *излучать сигналы небольшой мощности*, желательно не более 100 мВт (обычный сотовый телефон излучает сигналы мощностью от 600 мВт до 3 Вт).

Сегодня самой популярной технологией PAN является **Bluetooth**, которая обеспечивает взаимодействие 8 устройств в разделяемой среде диапазона 2,4 МГц со скоростью передачи данных до 723 Кбит/с.

Архитектура Bluetooth

Стандарт Bluetooth разработан группой Bluetooth SIG (Bluetooth Special Interest Group), которая была организована по инициативе компании Ericsson. Стандарт Bluetooth также адаптирован рабочей группой IEEE 802.15.1 в соответствии с общей структурой стандартов IEEE 802.

В технологии Bluetooth используется концепция **пикосети**. Название подчеркивает небольшую область покрытия, от 10 до 100 м, в зависимости от мощности излучения передатчика устройства. В пикосеть может входить до 255 устройств, но только 8 из них могут в каждый момент времени быть активными и обмениваться данными. Одно из устройств в пикосети является **главным**, остальные — **подчиненными** (рис. 12.21).

Активное подчиненное устройство может обмениваться данными только с главным устройством, прямой обмен между подчиненными устройствами невозможен. Все подчиненные устройства данной пикосети, кроме семи активных, должны находиться в режиме пониженного энергопотребления, в котором они только периодически прослушивают команду главного устройства для перехода в активное состояние.

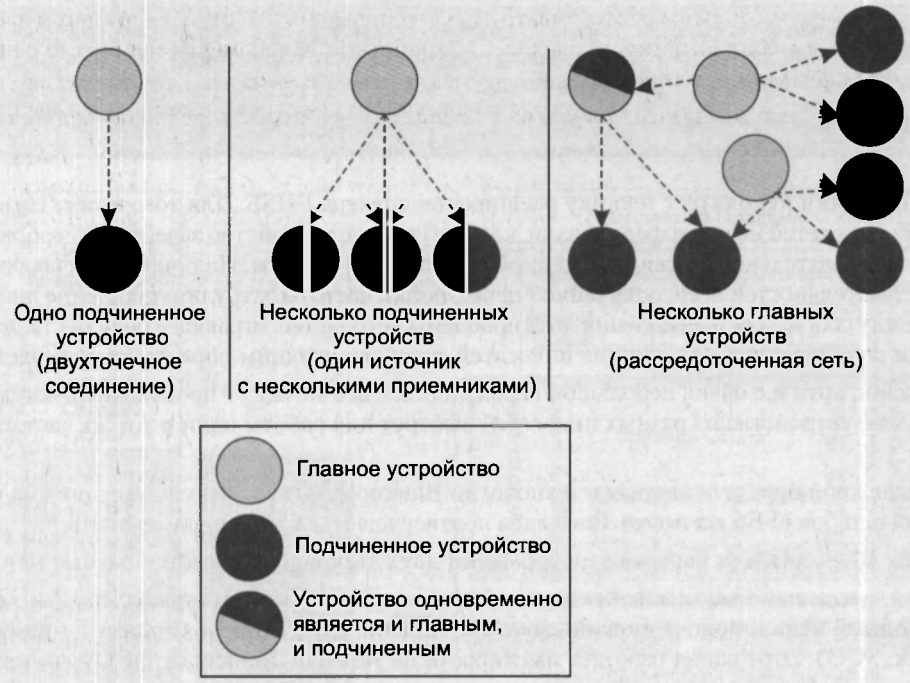


Рис. 12.21. Пикосеть и рассредоточенная сеть

Главное устройство отвечает за доступ к *разделяемой среде пикосети*, которая представляет собой нелицензируемые частоты диапазона 2,4 ГГц. Разделяемая среда передает данные со скоростью до 3 Мбит/с, но из-за накладных расходов на заголовки пакетов и смену частот полезная скорость передачи данных в среде не превышает 2,1 Мбит/с. Пропускная способность среды делится главным устройством между семью подчиненными устройствами на основе техники TDM.

Такая архитектура позволяет применять более простые протоколы в устройствах, выполняющих функции подчиненных (например, в радионаушниках), и отдает более сложные функции управления пикосетью компьютеру, который, скорее всего, и будет главным устройством этой сети.

Присоединение к пикосети происходит динамически. Главное устройство пикосети, используя процедуру опроса, собирает информацию об устройствах, которые попадают в зону его пикосети. После обнаружения нового устройства главное устройство проводит с ним переговоры. Если желание подчиненного устройства присоединиться к пикосети совпадает с решением главного устройства (подчиненное устройство прошло проверку аутентичности и оказалось в списке разрешенных устройств), то новое подчиненное устройство присоединяется к сети.

ПРИМЕЧАНИЕ

Безопасность сетей Bluetooth обеспечивается за счет аутентификации устройств и шифрования передаваемого трафика. Протоколы Bluetooth обеспечивают более высокий уровень защиты, чем протокол WEP стандарта IEEE 802.11.

Несколько пикосетей, которые обмениваются между собой данными, образуют **рассредоточенную сеть**. Взаимодействие в пределах рассредоточенной сети осуществляется за счет того, что один узел (называемый *мостом*) одновременно является членом нескольких пикосетей, причем этот узел может исполнять роль главного устройства одной пикосети и подчиненного устройства другой.

Сеть Bluetooth использует технику расширения спектра FHSS. Для того чтобы сигналы разных пикосетей не интерферировали, каждое главное устройство задействует *собственную* последовательность псевдослучайной перестройки частоты. Наличие различающихся последовательностей псевдослучайной перестройки частоты затрудняет общение пикосетей между собой. Для преодоления этой проблемы устройство, играющее роль моста, должно при подключении к каждой из пикосетей соответствующим образом менять частоту.

Коллизии, хотя и с очень небольшой вероятностью, все же могут происходить, когда два или более устройства из разных пикосетей выберут для работы один и тот же частотный канал.

Для надежной передачи данных в технологии Bluetooth может выполняться прямая коррекция ошибок (FEC), а получение кадра подтверждается с помощью квитанций.

В сетях Bluetooth для передачи информации двух типов используются разные методы.

- Для *чувствительного к задержкам трафика* (например, голоса) сеть поддерживает **синхронный канал, ориентированный на соединение** (Synchronous Connection-Oriented link, SCO). Этот канал работает на скорости 64 Кбит/с. Для канала SCO пропускная способность резервируется на все время соединения.
- Для *эластичного трафика* (например, компьютерных данных) используется работающий с переменной скоростью **асинхронный канал, не ориентированный на соединение** (Asynchronous Connection-Less link, ACL). Для канала ACL пропускная способность выделяется по запросу подчиненного устройства или по потребности главного устройства.

Стек протоколов Bluetooth

Bluetooth является законченной оригинальной технологией, рассчитанной на самостоятельное применение в электронных персональных устройствах. Поэтому эта технология поддерживает полный стек протоколов, включая собственные прикладные протоколы. В этом заключается ее отличие от рассмотренных ранее технологий, таких как Ethernet или IEEE 802.11, которые лишь выполняют функции физического и канального уровней.

Создание для технологии Bluetooth собственных прикладных протоколов объясняется стремлением разработчиков реализовывать ее в разнообразных простых устройствах, которым не под силу, да и не к чему, поддерживать стек протоколов TCP/IP. Кстати, технология Bluetooth появилась в результате попыток разработать стандарт для взаимодействия мобильного телефона с беспроводными наушниками. Понятно, что для решения такой простой задачи не нужен ни протокол передачи файлов (FTP), ни протокол передачи гипертекста (HTTP). В результате для технологии Bluetooth был создан оригинальный стек протоколов, в дополнение к которому появилось большое количество профилей.

Стек протоколов Bluetooth постоянно совершенствуется. Версия 1.0 стандартов стека была принята в 1999 году, версия 1.2 — в 2003, версия 2.0 — в 2004, версия 2.1 — в 2007, а версия 3.0 — в апреле 2009 года.

Профили определяют конкретный набор протоколов для решения той или иной задачи. Например, существует профиль для взаимодействия компьютера или мобильного телефона с беспроводными наушниками. Имеется также профиль для тех устройств, которые могут передавать файлы (наушникам он, скорее всего, не потребуется, хотя будущее предвидеть сложно), профиль эмуляции последовательного порта RS-232 и т. д.

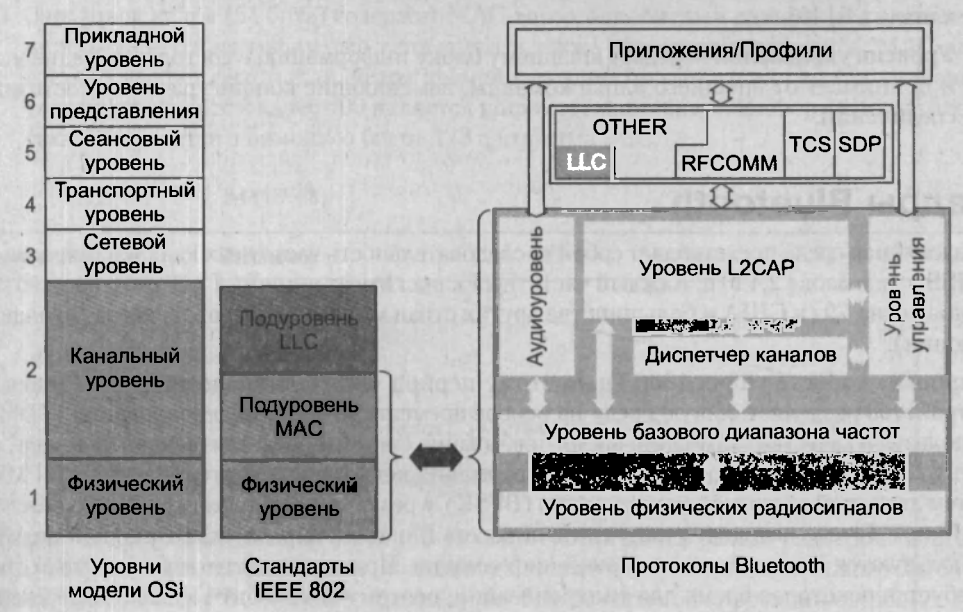


Рис. 12.22. Соответствие протоколов Bluetooth модели OSI и стандартам IEEE 802

При приведении стандартов Bluetooth в соответствие с архитектурой стандартов IEEE 802 рабочая группа 802.15.1 ограничилась только так называемыми протоколами *ядра Bluetooth*, которые соответствуют функциям физического уровня и уровня MAC (рис. 12.22).

- **Уровень физических радиосигналов** описывает частоты и мощности сигналов, используемых для передачи информации.
- **Уровень базового диапазона частот** отвечает за организацию каналов передачи данных в радиосреде. В его обязанности входят выбор последовательности псевдослучайной перестройки частоты, синхронизация устройств в пикосети, формирование и передача кадров по установленным каналам SCO и ACL. Кадр Bluetooth имеет переменную длину, поле данных может содержать от 0 до 2744 бит (343 байт). Для передачи голоса используются кадры фиксированного размера с полем данных 240 бит (30 байт).
- **Диспетчер каналов** отвечает за аутентификацию устройств и шифрование трафика, а также управляет статусом устройств, то есть может сделать подчиненное устройство главным, и наоборот.
- **Уровень протокола адаптации для управления логическим каналом** (Logical Link Control Adaptation Protocol, L2CAP) является верхним уровнем протоколов ядра Bluetooth. Этот протокол используется только в тех случаях, когда устройство передает данные; голосовой трафик обходит этот протокол и обращается непосредственно

к уровню базового диапазона частот. Уровень L2CAP принимает от протоколов верхнего уровня сегменты данных размером до 64 Кбайт и делит их на небольшие кадры для уровня базового диапазона частот. При приеме уровень L2CAP собирает кадры в исходный сегмент и передает протоколу верхнего уровня.

- **Аудиоуровень** обеспечивает передачу голоса по каналам SCO. На этом уровне применяется импульсно-кодовая модуляция (PCM), что определяет скорость голосового канала в 64 Кбит/с.
- **Уровень управления** передает внешнему блоку информацию о состоянии соединений и принимает от внешнего блока команды, изменяющие конфигурацию и состояние соединений.

Кадры Bluetooth

Разделяемая среда представляет собой последовательность частотных каналов технологии FHSS в диапазоне 2,4 ГГц. Каждый частотный канал имеет ширину 1 МГц, количество каналов равно 79 (в США и большинстве других стран мира) или 23 (в Испании, Франции, Японии).

Чиповая скорость равна 1600 Гц, поэтому период чипа составляет 625 мкс. Главное устройство разделяет общую среду на основе временного мультиплексирования (TDM), используя в качестве тайм-слота время пребывания системы на одном частотном канале, то есть 625 мкс. В версии протоколов 1.0 информация кодируется с тактовой частотой 1 МГц путем двоичной частотной манипуляции (BFSK), в результате битовая скорость составляет 1 Мбит/с. В течение одного тайм-слота пикосеть Bluetooth передает 625 бит, но не все они используются для передачи полезной информации. При смене частоты устройствам сети требуется некоторое время для синхронизации, поэтому из 625 бит только 366 передают кадр данных.

В версии 2.0 был введен режим **улучшенной скорости передачи данных** (Enhanced Data Rate, EDR), в котором для кодирования данных используется комбинация методов частотной (BFSK) и фазовой (PSK) модуляции; за счет этого удалось повысить битовую скорость до 3 Мбит/с, а полезную скорость передачи данных — до 2,1 Мбит/с. Режим EDR дополняет основной режим передачи данных со скоростью 1 Мбит/с.

Кадр данных может занимать 1, 3 или 5 слотов. В том случае, когда кадр занимает больше одного слота, частота канала остается неизменной в течение всего времени передачи кадра. В этом случае накладные расходы на синхронизацию меньше, так что размер кадра, состоящего, например, из 5 последовательных слотов, равен 2870 бит (с полем данных до 2744 бит).

ВНИМАНИЕ

Составными могут быть только кадры данных (то есть кадры канала ACL), а кадры, переносящие голос (кадры канала SCO), всегда состоят из одного слота.

Рассмотрим формат кадра, состоящего из одного слота — 366 бит (рис. 12.23):

- **Поле данных** занимает 240 бит.
- **Код доступа** (72 бита) служит для идентификации пикосети. Каждое устройство Bluetooth имеет глобально уникальный 6-байтовый адрес, поэтому для идентифика-

ции пикосети требуется три младших байта уникального адреса главного устройства. Каждое устройство при формировании кадра помещает эти байты в поле кода доступа, дополняя их битами 1/3 для прямой коррекции ошибок (сокращение 1/3 говорит о том, что 1 бит информации преобразуется в 3 бита кода). Если главное или подчиненное устройство получает кадр, содержащий неверный код доступа, то оно отбрасывает этот кадр, считая, что он, скорее всего, получен из другой пикосети.

- **Заголовок кадра** (54 бита) содержит MAC-адрес, однобитный признак подтверждения приема кадра, идентификатор типа кадра, а также ряд других признаков. MAC-адрес состоит из трех битов и является временным адресом одного их семи подчиненных устройств, при этом адрес 000 является широковещательным. Информация заголовка также передается с помощью битов 1/3 алгоритма FEC.

Рис. 12.23. Формат кадра Bluetooth, состоящего из одного слота

Формат кадра, состоящего из 3-х или 5-ти слотов, отличается только размером поля данных. Информация, помещаемая в поле данных, может кодироваться с помощью битов 1/3 или 2/3 алгоритма FEC либо передаваться вообще без прямой коррекции ошибок.

Поиск и стыковка устройств Bluetooth

Устройство, поддерживающее технологию Bluetooth, обычно посылает периодические запросы на предмет обнаружения других устройств Bluetooth в зоне досягаемости. Если устройство Bluetooth получает такой запрос и оно сконфигурировано таким образом, чтобы отвечать на запросы, то в ответ устройство передает сведения о себе: имя и тип устройства, имя производителя, поддерживаемые сервисы.

Имя устройства конфигурируется в отличие от его уникального MAC-адреса, который дается производителем. Нужно отметить, что часто устройства выпускаются со сконфигурированными по умолчанию именами, соответствующими названию модели устройства, поэтому в сфере досягаемости вашего мобильного телефона может оказаться несколько других телефонов с одинаковыми именами Bluetooth, если их владельцы не дали им собственные имена.

После предварительного обмена информацией устройства Bluetooth могут начать так называемую процедуру стыковки (Pairing), если конфигурация устройств ее требует. Стыковка подразумевает установление безопасного соединения между устройствами (см. главу 24); безопасность в данном случае означает, что устройства доверяют друг другу, а данные между ними передаются в зашифрованном виде. Стыковка устройств Bluetooth требует введения в каждое из них одного и того же пароля, называемого также PIN-кодом

Bluetooth. Обычно устройство, получившее запрос на стыковку, просит пользователя ввести PIN-код. Устройства, успешно прошедшие процедуру стыковки, запоминают этот факт и устанавливают безопасное соединение автоматически всякий раз, когда оказываются в зоне досягаемости, при этом повторное введение PIN-кода пользователем не требуется.

Устройство сможет быть сконфигурировано пользователем или производителем таким образом, чтобы разрешать установление соединений с другими устройствами без процедуры стыковки.

Пример обмена данными в пикосети

Рассмотрим работу пикосети на примере. Пусть пикосеть состоит из главного и трех активных подчиненных устройств. Для простоты предположим, что все устройства используют кадры, занимающие один слот. На рис. 12.24 показано, каким образом главное устройство распределяет слоты между членами пикосети.

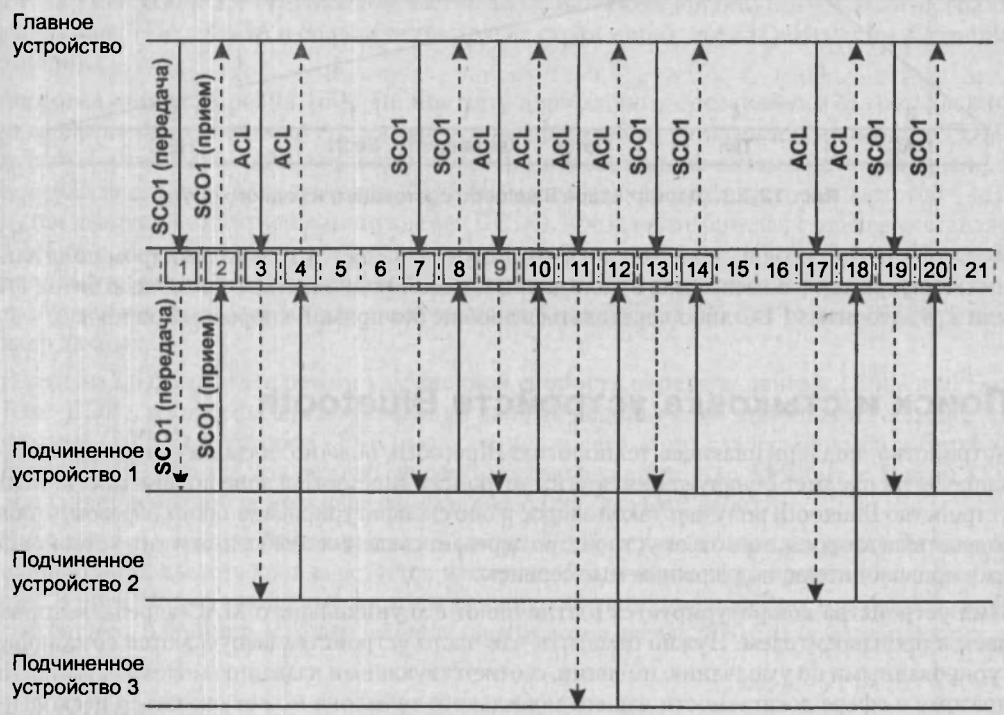


Рис. 12.24. Разделение среды

Для дуплексного обмена главное устройство всегда выделяет каждому каналу пару слотов: первый слот используется для передачи данных от главного устройства к подчиненному, а второй — в обратном направлении.

В примере, показанном на рисунке, существует один канал SCO между главным устройством и первым подчиненным устройством. Как мы уже знаем, каналам SCO всегда выделяется фиксированная часть пропускной способности среды, величина которой зависит

от того, каким образом будет использоваться метод прямой коррекции ошибок (FEC) голосовой информации.

- Если метод FEC не применяется, то для канала SCO выделяется каждая третья пара слотов, как это и показано на рисунке. Такое распределение слотов обеспечивает передачу потоков со скоростью 64 Кбит/с в каждом направлении. Убедимся в этом. Кодек PCM оцифровывает голос с частотой 8 кГц (период 125 мкс), представляя каждый замер одним байтом. Каждый кадр переносит 30 байт (240 бит), то есть 30 замеров. Кадры канала SCO в одном направлении повторяются через каждые 6 слотов, поэтому период повторения кадров равен $6 \times 625 = 3750$ мкс. Соответственно, скорость передачи данных в канале SCO (в одном направлении) равна $240/(3750 \times 10^{-6}) = 64$ Кбит/с.
- В том случае, когда используются биты 2/3 алгоритма FEC, то в поле данных кадра размещается не 30, а 20 замеров, поэтому для достижения скорости в 64 Кбит/с такому каналу SCO нужно выделять каждую вторую пару слотов.
- Наконец, биты 1/3 алгоритма FEC приводят к тому, что кадр переносит только 10 замеров голоса, так что такой канал занимает все слоты разделяемой среды.

Приведенные расчеты показывают, что в пикосети могут одновременно существовать не более трех каналов SCO (возможно, соединяющих с разными подчиненными устройствами), причем только тогда, когда канал не использует алгоритм FEC для снижения доли битовых ошибок. Прямая коррекция ошибок уменьшает число каналов SCO до двух или даже одного.

Оставшаяся от каналов SCO пропускная способность служит для передачи асинхронных данных. Для этого в пикосети имеется канал ACL. Этот канал соединяет один источник (главное устройство) с несколькими приемниками (все подчиненные устройства пикосети). Его не нужно устанавливать, он существует всегда.

Потребности подчиненных устройств в передаче асинхронных данных главное устройство узнает путем их периодического опроса. Для этого оно использует служебный кадр с MAC-адресом устройства. Если у главного устройства есть данные для этого подчиненного устройства, то оно может совместить передачу данных с опросом в одном кадре.

На рис. 12.24 показано, что главное устройство использовало слоты 3 и 4 для обмена кадрами со вторым подчиненным устройством, слоты 9 и 10 — для обмена с первым подчиненным устройством и слоты 11 и 12 — для обмена с третьим подчиненным устройством. Метод опроса исключает коллизии при доступе к каналу ACL, но скорость доступа к этому каналу для каждого отдельного устройства не определена, она зависит от количества устройств, которые хотят передавать асинхронные данные.

Таким образом, в сети Bluetooth совмещаются приемы коммутации каналов (для каналов SCO) и коммутации пакетов (для канала ACL).

В том случае, когда каналы SCO в сети не используются, вся пропускная способность среды отводится каналу ACL. При наличии кадров, состоящих из 5-ти слотов, максимальная скорость передачи данных при битовой скорости 1 Мбит/с составляет 432,6 Кбит/с в каждом направлении (без прямой коррекции ошибок). Допустимо также несимметричное деление пропускной способности канала ACL, тогда максимальная скорость достигает 723,2 Кбит/с в одном направлении при скорости 57,6 Кбит/с в обратном. Не нужно забывать, что это — суммарные скорости передачи данных в канале ACL, а не скорости потоков данных отдельных устройств. Когда несколько устройств используют канал, скорость делится между всеми устройствами.

Новые свойства Bluetooth

В последних версиях стандартов Bluetooth были анонсированы некоторые нововведения, одно из которых — повышение скорости передачи данных в режиме EDR до 3 Мбит/с — мы уже упомянули. Далее перечислены другие наиболее важные новые свойства этой технологии.

- *Пониженная скорость обмена в ждущем режиме.* Это свойство заключается в снижении частоты обмена служебными сообщениями keepalive («работоспособен»), которыми узлы поддерживают соединение в открытом состоянии при отсутствии пользовательских данных для передачи, с нескольких сообщений в секунду до одного сообщения раз в 5 или 10 секунд. Такой режим позволяет увеличить время работы батарей портативных устройств в 3–10 раз. Свойство введено в версии 2.1.
- *Безопасная простая стыковка* (secure simple pairing) позволяет ускорить процедуру стыковки и в то же время предлагает более высокую степень защиты соединений. Свойство введено в версии 2.1.
- *Использование технологии NFC* (Near Field Communication — связь ближнего радиуса действия) для автоматической стыковки устройств. NFC — это новая технология, разработанная для беспроводного взаимодействия устройств на расстояниях в 10–20 см. При обнаружении сигналов устройства с интерфейсами NFC автоматически устанавливают соединение. Устройства Bluetooth могут использовать технологию NFC для автоматического обнаружения при приближении их друг к другу в ходе стыковки и обмена информацией. Это свойство является частью упомянутой ранее процедуры безопасной простой стыковки, оно также введено в версии 2.1 Bluetooth.
- *Альтернативные MAC-уровень и физический уровень.* При необходимости передачи большого объема данных устройство Bluetooth может переключиться на соединение, использующее отличную от Bluetooth технологию передачи данных. В версии 3.0 протоколов Bluetooth как возможная альтернатива определены пока только технологии 802.11, но в будущем могут быть стандартизованы и другие технологии. Первоначальное взаимодействие устройств всегда должно производиться на основе технологии Bluetooth.
- *Bluetooth с низким энергопотреблением (Bluetooth low energy).* В апреле 2009 года группа Bluetooth SIG объявила о совершенно новом дополнительном стеке протоколов под названием Bluetooth low energy. Этот стек разрабатывался группой Bluetooth совместно с компанией Nokia и был первоначально известен под названием Wibree. Протоколы Bluetooth low energy предназначены для устройств, батареи которых должны иметь примерно годичный срок действия; это могут быть, например, наручные часы или медицинские приборы.

Выводы

Локальные сети на разделяемой среде представляют собой наиболее простой и дешевый в реализации тип локальных сетей. Основной недостаток разделяемых локальных сетей состоит в плохой масштабируемости, так как при увеличении числа узлов сети снижается доля пропускной способности, приходящаяся на каждый узел.

Уровень MAC отвечает за доступ к разделяемой среде и отправку через нее кадров.

Протокол LLC обеспечивает для протоколов верхних уровней нужное качество транспортных услуг, передавая кадры либо дейтаграммным способом, либо с помощью процедур с установлением соединения и восстановлением кадров.

В технологии Ethernet на разделяемой среде применяется случайный метод доступа CSMA/CD, который очень прост в реализации.

Коллизия — это ситуация, когда две станции одновременно пытаются передать кадр данных через общую среду. Наличие коллизий — это неотъемлемое свойство сетей Ethernet, являющееся следствием принятого случайного метода доступа.

В зависимости от типа физической среды стандарт IEEE 802.3 определяет различные спецификации Ethernet со скоростью 10 Мбит/с: 10Base-5, 10Base-2, 10Base-T, FOIRL, 10Base-FL, 10Base-FB.

В сетях Token Ring используется детерминированный метод доступа с передачей токена. Логической топологией сетей Token Ring является кольцо, физической — звезда. За счет кольцевой топологии технология Token Ring отчасти обеспечивает отказоустойчивость.

В технологии FDDI в качестве физической среды впервые был использован волоконно-оптический кабель и достигнута скорость 100 Мбит/с. Высокая степень отказоустойчивости обеспечивается за счет применения двойного оптоволоконного кольца.

Стандарты IEEE 802.11 являются основными стандартами беспроводных локальных сетей. Существует несколько вариантов спецификаций физического уровня 802.11, отличающихся диапазоном используемых частот (2,4 и 5 ГГц), а также методом кодирования (FHSS, DSSS, OFDM).

Метод доступа 802.11 является комбинацией случайного метода доступа с предотвращением коллизий (DCF) и централизованного детерминированного метода доступа с опросом (PCF). Гибкое применение режимов DCF и PCF позволяет обеспечить поддержку показателей QoS для синхронного и асинхронного трафиков.

Персональные сети (PAN) предназначены для взаимодействия принадлежащих одному владельцу устройств на небольшом расстоянии, обычно в радиусе от 10 до 100 м. Персональные сети должны обеспечивать как фиксированный доступ, например, в пределах дома, так и мобильный, когда владелец устройств перемещается вместе с ними между помещениями или городами.

Сегодня самой популярной технологией PAN является Bluetooth, в которой используется концепция пикосети, объединяющей до 255 устройств, но только 8 из них могут в каждый момент времени быть активными.

Для чувствительного к задержкам трафика сеть Bluetooth поддерживает синхронные каналы, ориентированные на соединение (SCO), а для эластичного — асинхронные каналы, не ориентированные на соединение (ACL).

Вопросы и задания

1. Выберите утверждения, корректно описывающие особенности метода доступа технологии Ethernet:
   а) узел обязан «прослушивать» разделяемую среду;
   б) узел может передать свой кадр в разделяемую среду в любой момент времени независимо от того, занята среда или нет;
   в) узел ожидает подтверждения приема переданного кадра от узла назначения в течение некоторого времени, а в случае истечения этого времени повторяет передачу;
   г) если в течение времени передачи кадра коллизия не произошла, то кадр считается переданным успешно.
2. Почему протоколы канального уровня технологий глобальных сетей не делятся на подуровни MAC и LLC?

3. Какие функции выполняет уровень LLC?
   - а) управляет доступом к логическому интерфейсу;
   - б) поддерживает интерфейс с вышележащим уровнем;
   - в) обеспечивает передачу кадра с заданным уровнем надежности;
   - г) разрешает коллизии.
4. При увеличении длины разделяемого сегмента Ethernet и расстояний между подключенными к нему узлами, но при сохранении числа подключенных к сегменту узлов, что будет с вероятностью коллизий? Варианты ответов:

   а) понизится; б) повысится; в) не изменится.
5. В чем состоят функции преамбулы и начального ограничителя кадра в стандарте Ethernet?
6. Чем объясняется, что минимальный размер поля данных кадра Ethernet выбран равным 46 байт? Варианты ответов:

   а) для предотвращения монопольного захвата среды узлом;

   б) для устойчивого распознавания коллизий;

   в) для сокращения накладных расходов.
7. Почему сети 10Base-T и 10Base-F вытеснили в свое время сети Ethernet на коаксиальном кабеле?
8. Какова была скорость передачи пользовательских данных в сети Ethernet 10Base-T при передаче файла между сервером и клиентом, если средняя длина кадров при этом равнялась 920 байт с учетом полей заголовков, но без учета преамбулы, а кадры передавались сервером с минимально возможным межкадровым интервалом и без коллизий?
9. К какому типу относится MAC-адрес 01:80:C2:00:00:08? Варианты ответов:

   а) групповой; б) индивидуальный; в) локальный; г) централизованный.
10. Как скорость передачи данных технологии Ethernet на разделяемой среде влияет на максимальный диаметр сети? Варианты ответов:

    а) чем выше скорость передачи, тем меньше максимальный диаметр сети;

    б) чем выше скорость передачи, тем больше максимальный диаметр сети;

    в) не влияет.
11. Какое максимальное время должно пройти до того момента, когда кадр будет отброшен адаптером Ethernet из-за постоянных коллизий при передаче?
12. Что произойдет, если соединить кабелем два порта концентратора Ethernet?
13. Какой механизм предотвращает монопольное использование кольца Token Ring каким-либо ее узлом? Варианты ответов:

    а) система приоритетов кадров;

    б) таймер времени удержания токена;

    в) кольцевая топология сети.
14. Какой элемент обеспечивает отказоустойчивость сети Token Ring? Варианты ответов:

    а) сетевой адаптер; б) вторичное кольцо; в) повторитель.

15. Какой элемент делает отказоустойчивость сети FDDI выше, чем сети Token Ring? Варианты ответов:

    а) сетевой адаптер; б) вторичное кольцо; в) повторитель.

16. К чему приводит наличие скрытого терминала в сети IEEE 802.11? Варианты ответов:

    а) к нарушению связности сети;

    б) к повышению уровня помех в радиосреде;

    в) к более частому возникновению коллизий.

17. Каким образом обнаруживает коллизии уровень MAC в сетях 802.11?

18. Может ли станция сети 802.11 передать кадр другой входящей в ту же сеть BSS станции не непосредственно, а через точку доступа?

19. Из каких соображений выбирается длительность слота в режиме DCF? Варианты ответов:

    а) длительность слота должна превосходить время распространения сигнала между любыми станциями сети;

    б) длительность слота не должна превосходить время передачи кадра максимальной длины;

    в) длительность слота должна превосходить время распространения сигнала между любыми станциями сети плюс время, затрачиваемое станцией на распознавание занятости среды.

20. За счет чего режим PCF всегда имеет приоритет перед режимом DCF?

21. Каким образом пикосети Bluetooth объединяются в рассредоточенную сеть? Варианты ответов:

    а) с помощью маршрутизатора;

    б) с помощью коммутатора;

    в) с помощью узла, являющегося членом нескольких пикосетей.

ГЛАВА 13 Коммутируемые сети Ethernet

Современные коммутаторы Ethernet являются наследниками мостов локальных сетей, которые широко использовались в сетях Ethernet и Token Ring на разделяемой среде. Более того, коммутаторы Ethernet по-прежнему функционально очень близки к вышедшим из употребления мостам, так как базовый алгоритм работы коммутатора и моста является одним и тем же алгоритмом и определяется одним и тем же стандартом IEEE 802.1D. По традиции во всех новых стандартах IEEE, описывающих свойства коммутаторов, употребляется термин «коммутатор», а не «мост». Основное отличие коммутатора от моста состоит в большем количестве портов (мост, как правило, имел два порта, что и послужило поводом для его названия — мост между двумя сегментами) и более высокой производительности.

Коммутаторы являются сегодня основным типом коммуникационных устройств, применяемых для построения локальных сетей. Коммутаторы отличаются внутренней архитектурой и конструктивным исполнением.

Мост как предшественник и функциональный аналог коммутатора

Логическая структуризация сетей и мосты

Мост локальной сети (LAN bridge), или просто **мост**, появился как средство построения крупных локальных сетей на разделяемой среде, так как в рамках того, что в стандартах сетей на разделяемой среде называется сетью, построить действительно крупную сеть практически невозможно, поскольку такая сеть подразумевает существование единой разделяемой среды.

В сети Ethernet требование использовать единую разделяемую среду приводит к нескольким очень жестким ограничениям:

- общий диаметр сети не может быть больше 2500 м;
- количество узлов не может превышать 1024 (для сетей Ethernet на коаксиале это ограничение еще жестче).

На практике из-за главной проблемы разделяемой среды — дефицита пропускной способности — количество узлов даже в сетях 10Base-T и 10Base-F никогда не приближается к 1024.

Процессы, происходящие в локальных сетях на разделяемой среде, качественно могут быть описаны моделями массового обслуживания, в частности моделью M/M/1, рассмотренной в главе 7. Разделяемая среда соответствует обслуживающему устройству этой модели, а кадры, генерируемые каждым компьютером сети, — заявкам на обслуживание. Очередь заявок в действительности распределяется по компьютерам сети, где кадры ожидают своей очереди на использование среды.

Хотя модель M/M/1 не может адекватно отразить многие особенности локальных сетей на разделяемой среде, например коллизии, возникающие в Ethernet, она хорошо иллюстрирует качественную картину зависимости задержек доступа к среде от коэффициента использования среды. На рис. 13.1 показаны зависимости этого типа, полученные для сетей Ethernet, Token Ring и FDDI путем имитационного моделирования.

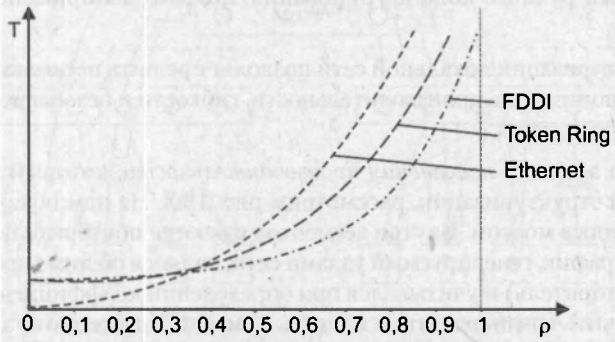


Рис. 13.1. Задержки доступа к среде передачи данных для технологий Ethernet, Token Ring и FDDI

Как видно из рисунка, всем технологиям присуща качественно одинаковая картина экспоненциального роста величины задержек доступа при увеличении коэффициента использования сети. Однако их отличает порог, при котором наступает резкий перелом в поведении сети, когда почти прямолинейная зависимость переходит в крутую экспоненциальную. Для всего семейства технологий Ethernet — это 30–50 % (сказывается эффект коллизий), для технологии Token Ring — 60 %, а для технологии FDDI — 70–80 %.

Количество узлов, при которых коэффициент использования сети начинает приближаться к опасной границе, зависит от типа функционирующих в узлах приложений. Для сетей Ethernet со скоростью 10 Мбит/с считалось, что 30 узлов — это вполне приемлемое число для одного разделяемого сегмента, так что для построения крупной сети нужны были принципиально новые решения.

Ограничения, возникающие из-за использования единой разделяемой среды, можно преодолеть, выполнив *логическую структуризацию сети*, то есть сегментировав единую разделяемую среду на несколько и соединив полученные сегменты сети некоторым коммуникационным устройством, которое не передает данные побитно, как повторитель, а буферизует кадры и передает их затем в тот или иной сегмент (или сегменты) в зависимости от адреса назначения кадра (рис. 13.2). То есть такие сегменты работают в соответствии с обобщенным алгоритмом коммутации, рассмотренном в главе 2.

Нужно отличать логическую структуризацию от физической. Концентраторы стандарта 10Base-T позволяют построить сеть, состоящую из нескольких сегментов кабеля на витой паре, но это — физическая структуризация, так как логически все эти сегменты представляют собой единую разделяемую среду.

Мост долгое время был основным типом устройств, которые использовались для логической структуризации локальных сетей. Сейчас мосты заменили коммутаторы, но так как алгоритм их работы повторяет алгоритм работы моста, результаты их применения имеют ту же природу, они только усиливаются за счет гораздо более высокой производительности коммутаторов.

Помимо мостов/коммутаторов для структуризации локальных сетей можно использовать маршрутизаторы, но они являются более сложными и дорогими устройствами, к тому же всегда требующими ручного конфигурирования, поэтому их применение в локальных сетях ограничено.

Логическая структуризация локальной сети позволяет решить несколько задач, основные из которых — это повышение производительности, гибкости и безопасности, а также улучшение управляемости сети.

Для иллюстрации эффекта *повышения производительности*, который является главной целью логической структуризации, рассмотрим рис. 13.3. На нем показаны два сегмента Ethernet, соединенные мостом. Внутри сегментов имеются повторители. До деления сети на сегменты весь трафик, генерируемый узлами сети, являлся общим (представим, что вместо моста был повторитель) и учитывался при определении коэффициента использования сети. Если обозначить среднюю интенсивность трафика, идущего от узла i к узлу j, через Cij, то суммарный трафик, который должна была передавать сеть до деления на сегменты, равен $C\Sigma = \Sigma Cij$ (считаем, что суммирование проводится по всем узлам).

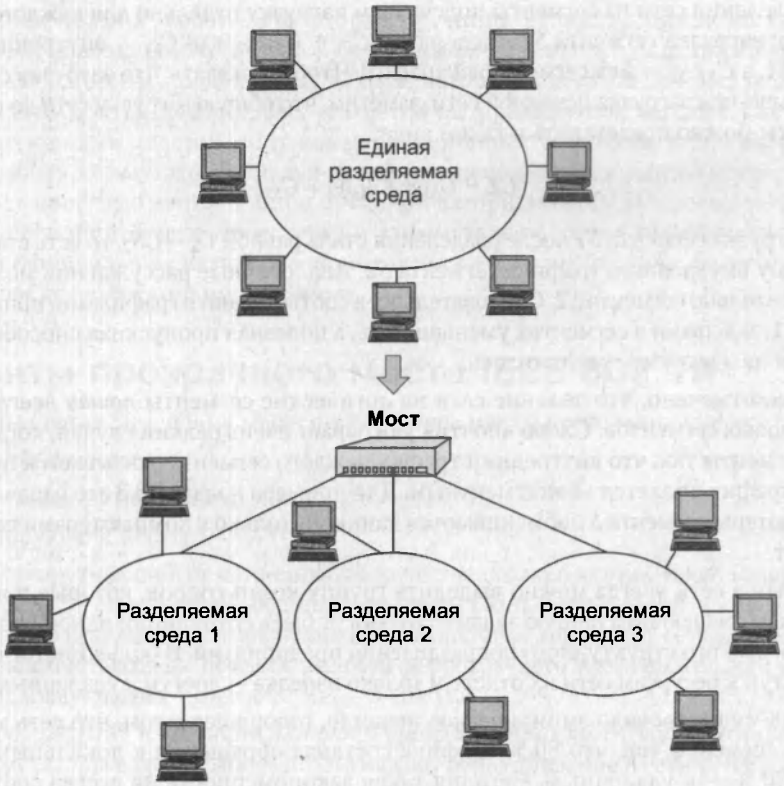


Рис. 13.2. Логическая структуризация сети

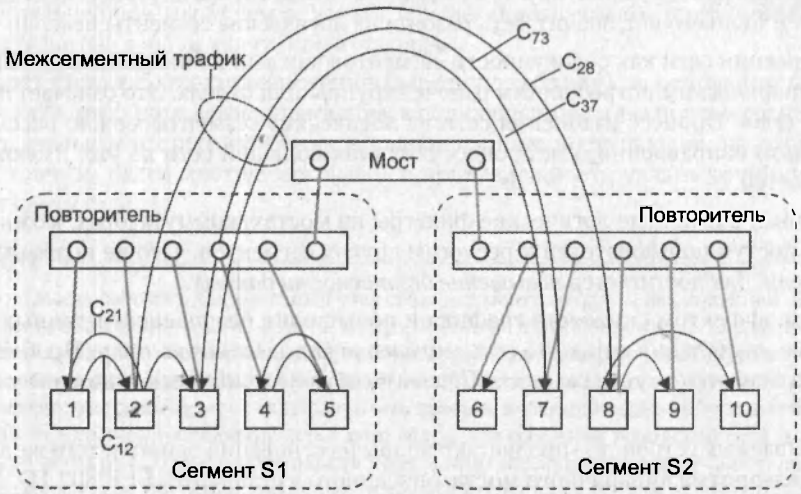


Рис. 13.3. Изменение загрузки при делении сети на сегменты

После разделения сети на сегменты подсчитаем нагрузку отдельно для каждого сегмента. Например, нагрузка сегмента $S1$ стала равна $C_{S1} + C_{S1\text{-}S2}$, где C_{S1} — внутренний трафик сегмента $S1$, а $C_{S1\text{-}S2}$ — межсегментный трафик. Чтобы показать, что загрузка сегмента $S1$ стала меньше, чем загрузка исходной сети, заметим, что общую загрузку сети до разделения на сегменты можно представить в таком виде:

$$C\Sigma = C_{S1} + C_{S1\text{-}S2} + C_{S2}.$$

Значит, загрузка сегмента $S1$ после разделения стала равной $C_{\Sigma} - CS_2$, то есть стала меньше на величину внутреннего трафика сегмента $S2$. Аналогичные рассуждения можно повторить относительно сегмента $S2$. Следовательно, в соответствии с графиками, приведенными на рис. 13.1, задержки в сегментах уменьшились, а полезная пропускная способность, приходящаяся на один узел, увеличилась.

Ранее было отмечено, что деление сети на логические сегменты *почти* всегда снижает загрузку новых сегментов. Слово «почти» учитывает очень редкий случай, когда сеть разбита на сегменты так, что внутренний трафик каждого сегмента оказывается нулевым, то есть весь трафик является межсегментным. Для примера на рис. 13.3 это означало бы, что все компьютеры сегмента $S1$ обмениваются данными только с компьютерами сегмента $S2$, и наоборот.

На практике в сети всегда можно выделить группу компьютеров, которые принадлежат сотрудникам, решающим общую задачу. Это могут быть сотрудники одной рабочей группы, отдела, другого структурного подразделения предприятия. В большинстве случаев им нужен доступ к ресурсам сети их отдела и только изредка — доступ к удаленным ресурсам.

В 80-е годы существовало эмпирическое правило, говорящее о том, что сеть можно разделить на сегменты так, что 80 % трафика составят обращения к локальным ресурсам и только 20 % — к удаленным. Сегодня такая закономерность не всегда соответствует действительности, она может трансформироваться в правило 50 на 50 % и даже 20 на 80 % (например, большая часть обращений направлена к ресурсам Интернета или к централизованным серверам предприятия). Тем не менее в любом случае внутрисегментный трафик существует. Если его нет, значит, сеть разбита на логические сегменты неверно.

При построении сети как совокупности сегментов каждый из них может быть адаптирован к специфическим потребностям рабочей группы или отдела. Это означает *повышение гибкости сети*. Процесс разбиения сети на логические сегменты можно рассматривать и в обратном направлении, как процесс создания большой сети из уже имеющихся небольших сетей.

Устанавливая различные логические фильтры на мостах/коммутаторах, можно контролировать доступ пользователей к ресурсам других сегментов, чего не позволяют делать повторители. Так достигается *повышение безопасности данных*.

Побочным эффектом снижения трафика и повышения безопасности данных является упрощение управления сетью, то есть *улучшение управляемости сети*. Проблемы очень часто локализуются внутри сегмента. Сегменты образуют логические домены управления сетью.

Оба описываемых устройства продвигают кадры на основании одного и того же алгоритма, а именно **алгоритма прозрачного моста**, описанного в стандарте IEEE 802.1D.

Этот стандарт, разработанный задолго до появления первого коммутатора, описывал работу *моста*, поэтому совершенно естественно, что в его названии и содержании

используется термин «мост». Некоторая путаница возникла, когда на свет появились первые модели коммутаторов — они выполняли тот же описанный в стандарте IEEE 802.1D алгоритм продвижения кадров, который с десяток лет был отработан мостами. И хотя мосты, для которых алгоритм был разработан, сегодня уже относятся к практически «вымершему» виду коммуникационных устройств, в стандартах, описывающих работу коммутатора, следуя традиции, используют термин «мост». Мы же не будем столь консервативными и при описании алгоритмов 802.1D в следующем разделе позволим себе иногда указывать термин «коммутатор», кроме тех случаев, когда речь пойдет об официальном названии стандарта или когда необходимо будет подчеркнуть разницу между двумя типами устройств.

Алгоритм прозрачного моста IEEE 802.1D

В локальных сетях 80-х и 90-х годов применялись мосты нескольких типов:

- прозрачные мосты;
- мосты с маршрутизацией от источника;
- транслирующие мосты.

Мосты с маршрутизацией от источника применялись только в сетях Token Ring, а транслирующие мосты были способны соединять сегменты разных технологий, например Ethernet и Token Ring. В результате исчезновения всех технологий локальных сетей, кроме Ethernet, оба этих типа мостов также исчезли, а алгоритм прозрачного моста выжил, найдя свое применение в коммутаторах Ethernet.

Слово «прозрачный» в названии *алгоритм прозрачного моста* отражает тот факт, что мосты и коммутаторы в своей работе не учитывают существование в сети сетевых адаптеров конечных узлов, концентраторов и повторителей. В то же время и перечисленные сетевые устройства функционируют, «не замечая» присутствия в сети мостов и коммутаторов.

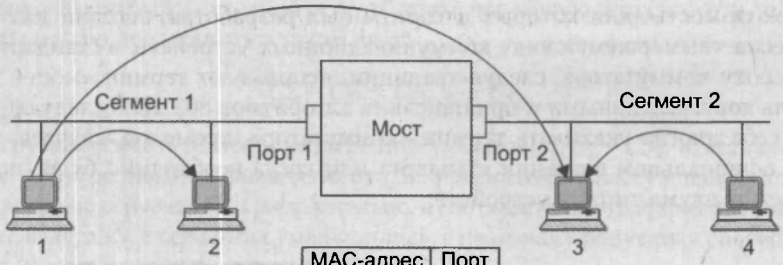
Так как алгоритм прозрачного моста остался единственным актуальным алгоритмом мостов, то в дальнейшем мы будем опускать термин «прозрачный», подразумевая именно этот тип алгоритма работы моста/коммутатора.

Мост строит свою таблицу продвижения (адресную таблицу) на основании пассивного наблюдения за трафиком, циркулирующим в подключенных к его портам сегментах. При этом мост учитывает адреса источников кадров данных, поступающих на его порты. По адресу источника кадра мост делает вывод о принадлежности узла-источника тому или иному сегменту сети.

ВНИМАНИЕ

Каждый порт моста работает, как конечный узел своего сегмента, за одним исключением — порт моста может не иметь собственного MAC-адреса. Порты мостов не нуждаются в адресах для продвижения кадров, так как они работают в режиме неразборчивого захвата кадров, когда все поступающие на порт кадры, независимо от их адреса назначения, запоминаются на время в буферной памяти. Работая в неразборчивом режиме, мост «слушает» весь трафик, передаваемый в присоединенных к нему сегментах, и использует проходящие через него кадры для изучения топологии сети и построения таблицы продвижения. В том случае, когда порт моста/коммутатора имеет собственный MAC-адрес, он используется для целей, отличных от продвижения кадров, чаще всего — для удаленного управления портом; в этом случае порт представляет собой конечный узел сети, и кадры адресуются непосредственно ему.

Рассмотрим процесс автоматического создания таблицы продвижения моста и ее использования на примере простой сети, представленной на рис. 13.4.



MAC-адрес	Порт
1	1
2	1
3	2
4	2

Рис. 13.4. Принцип работы прозрачного моста/коммутатора

Мост соединяет два сетевых сегмента. Сегмент 1 составляют компьютеры, подключенные с помощью одного отрезка коаксиального кабеля к порту 1 моста, а сегмент 2 — компьютеры, подключенные с помощью другого отрезка коаксиального кабеля к порту 2 моста. В исходном состоянии мост не знает о том, компьютеры с какими MAC-адресами подключены к каждому из его портов. В этой ситуации мост просто передает любой захваченный и буферизованный кадр на *все* свои порты за исключением того порта, от которого этот кадр получен. В нашем примере у моста только два порта, поэтому он передает кадры с порта 1 на порт 2, и наоборот. Отличие работы моста в этом режиме от повторителя заключается в том, что он передает кадр, предварительно буферизуя его, а не бит за битом, как это делает повторитель. Буферизация разрывает логику работы всех сегментов как единой разделяемой среды. Когда мост собирается передать кадр с сегмента на сегмент, например с сегмента 1 на сегмент 2, он, как обычный конечный узел, пытается получить доступ к разделяемой среде сегмента 2 по правилам алгоритма доступа, в данном примере — по правилам алгоритма CSMA/CD.

Одновременно с передачей кадра на все порты мост изучает адрес источника кадра и делает запись о его принадлежности к тому или иному сегменту в своей **адресной таблице**. Эту таблицу также называют **таблицей фильтрации**, или **продвижения**. Например, получив на порт 1 кадр от компьютера 1, мост делает первую запись в своей адресной таблице:

MAC-адрес 1 — порт 1.

Эта запись означает, что компьютер, имеющий MAC-адрес 1, принадлежит сегменту, подключенному к порту 1 коммутатора. Если все четыре компьютера данной сети проявляют активность и посылают друг другу кадры, то скоро мост построит полную адресную таблицу сети, состоящую из 4-х записей — по одной записи на узел (см. рис. 13.4).

При каждом поступлении кадра на порт моста он, прежде всего, пытается найти адрес назначения кадра в адресной таблице. Продолжим рассмотрение действий моста на примере (см. рис. 13.4).

1. При получении кадра, направленного от компьютера 1 компьютеру 3, мост просматривает адресную таблицу на предмет совпадения адреса в какой-либо из ее записей

с адресом назначения — MAC-адресом 3. Запись с искомым адресом имеется в адресной таблице.

2. Мост выполняет второй этап анализа таблицы — проверяет, находятся ли компьютеры с адресами источника и назначения в одном сегменте. В примере компьютер 1 (MAC-адрес 1) и компьютер 3 (MAC-адрес 3) находятся в разных сегментах. Следовательно, мост выполняет операцию **продвижения** (forwarding) кадра — передает кадр на порт 2, ведущий в сегмент получателя, получает доступ к сегменту и передает туда кадр.
3. Если бы оказалось, что компьютеры принадлежали одному сегменту, то кадр просто был бы удален из буфера. Такая операция называется **фильтрацией** (filtering).
4. Если бы запись о MAC-адресе 3 отсутствовала в адресной таблице, то есть, другими словами, *адрес назначения был неизвестен* мосту, то он передал бы кадр на все свои порты, кроме порта — источника кадра, как и на начальной стадии процесса обучения.

Процесс обучения моста никогда не заканчивается и происходит одновременно с продвижением и фильтрацией кадров. Мост постоянно следит за адресами источника буферизуемых кадров, чтобы автоматически приспосабливаться к изменениям, происходящим в сети, — перемещениям компьютеров из одного сегмента сети в другой, отключению и появлению новых компьютеров.

Входы адресной таблицы могут быть динамическими, создаваемыми в процессе *самообучения* моста, и статическими, создаваемыми *вручную* администратором сети. **Статические записи** не имеют срока жизни, что дают администратору возможность влиять на работу моста, например ограничивая передачу кадров с определенными адресами из одного сегмента в другой.

Динамические записи имеют срок жизни — при создании или обновлении записи в адресной таблице с ней связывается отметка времени. По истечении определенного тайм-аута запись помечается как недействительная, если за это время мост не принял ни одного кадра с данным адресом в поле адреса источника. Это дает возможность мосту автоматически реагировать на перемещения компьютера из сегмента в сегмент — при его отключении от старого сегмента запись о принадлежности компьютера к этому сегменту со временем вычеркивается из адресной таблицы. После подключения компьютера к другому сегменту его кадры начнут попадать в буфер моста через другой порт, и в адресной таблице появится новая запись, соответствующая текущему состоянию сети.

Кадры с широковещательными MAC-адресами, как и кадры с неизвестными адресами назначения, передаются мостом на все его порты. Такой режим распространения кадров называется **затоплением сети** (flooding). Наличие мостов в сети не препятствует распространению широковещательных кадров по всем сегментам сети. Однако это является достоинством только тогда, когда широковещательный адрес выработан корректно работающим узлом.

Нередко в результате каких-либо программных или аппаратных сбоев протокол верхнего уровня или сетевой адаптер начинает работать некорректно, а именно постоянно с высокой интенсивностью генерировать кадры с широковещательным адресом. Мост в соответствии со своим алгоритмом передает ошибочный трафик во все сегменты. Такая ситуация называется **широковещательным штормом** (broadcast storm).

К сожалению, мосты не защищают сети от широковещательного шторма, во всяком случае, по умолчанию, как это делают маршрутизаторы (вы познакомитесь с этим свойством маршрутизаторов в части IV). Максимум, что может сделать администратор с помощью

коммутатора для борьбы с широковещательным штормом — установить для каждого порта моста предельно допустимую интенсивность передачи кадров с широковещательным адресом. Но при этом нужно точно знать, какая интенсивность является нормальной, а какая — ошибочной. При смене протоколов ситуация в сети может измениться, и то что вчера считалось ошибочным, сегодня может оказаться нормой.

На рис. 13.5 показана типичная структура моста. Функции доступа к среде при приеме и передаче кадров выполняют микросхемы MAC, которые идентичны микросхемам сетевого адаптера.

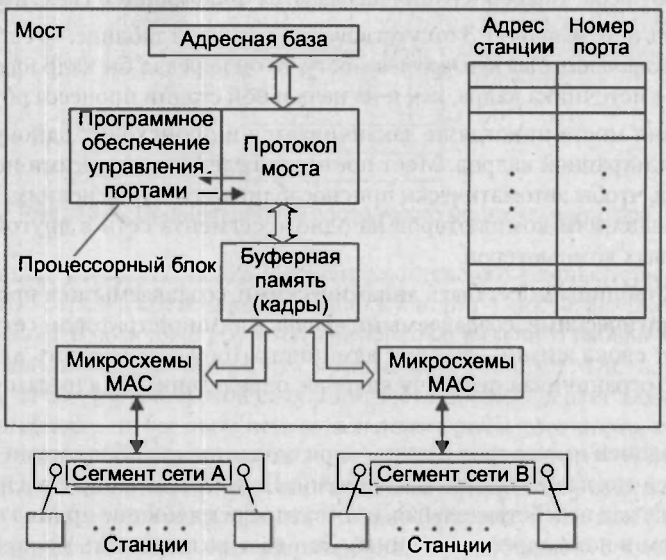


Рис. 13.5. Структура моста/коммутатора

Протокол, реализующий алгоритм коммутатора, располагается между уровнями MAC и LLC.

На рис. 13.6 показана копия экрана терминала с адресной таблицей моста.

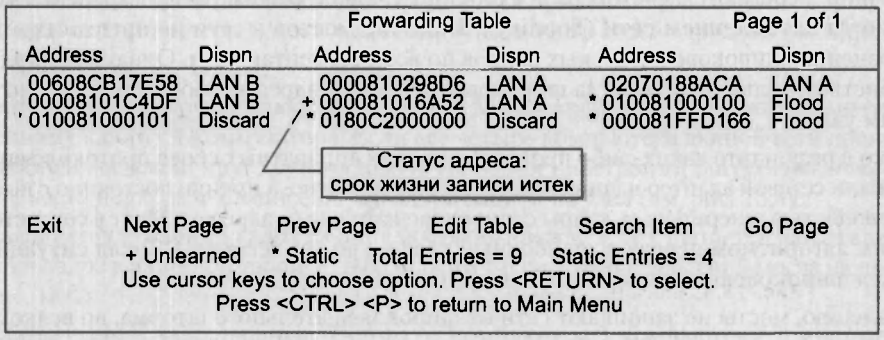


Рис. 13.6. Адресная таблица коммутатора

Из выводимой на экран адресной таблицы видно, что сеть состоит из двух сегментов — LAN *A* и LAN *B*. В сегменте LAN *A* имеются, по крайней мере, 3 станции, а в сегменте LAN *B* — 2 станции. Четыре адреса, помеченные звездочками, являются статическими, то есть назначенными администратором вручную. Адрес, помеченный плюсом, является динамическим адресом с истекшим сроком жизни.

Таблица имеет поле *Dispn* — «disposition» (это «распоряжение» мосту о том, какую операцию нужно проделать с кадром, имеющим данный адрес назначения). Обычно при автоматическом составлении таблицы в этом поле ставится условное обозначение порта назначения, но при ручном задании адреса в это поле можно внести нестандартную операцию обработки кадра. Например, операция *Flood* (затопление) заставляет мост распространять кадр в широковещательном режиме, несмотря на то что его адрес назначения не является широковещательным. Операция *Discard* (отбросить) говорит мосту, что кадр с таким адресом не нужно передавать на порт назначения. Вообще говоря, операции, задаваемые в поле *Dispn*, определяют особые условия фильтрации кадров, дополняющие стандартные условия их распространения. Такие условия обычно называют **пользовательскими фильтрами**, мы их рассмотрим немного позже в разделе «Фильтрация трафика» главы 14.

Топологические ограничения при применении мостов в локальных сетях

Серьезным ограничением функциональных возможностей мостов и коммутаторов является отсутствие поддержки петлеобразных конфигураций сети.

Рассмотрим это ограничение на примере сети, показанной на рис. 13.7.

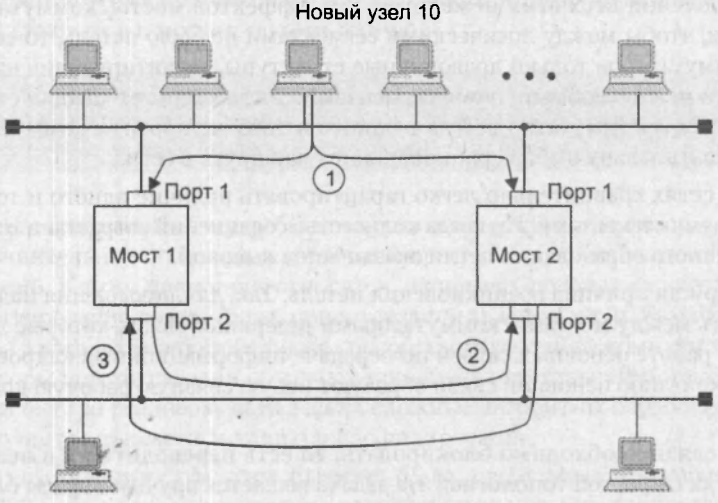


Рис. 13.7. Влияние замкнутых маршрутов на работу коммутаторов

Два сегмента Ethernet параллельно соединены двумя мостами, так что образовалась петля. Пусть новая станция с MAC-адресом 123 впервые начинает работу в данной сети. Обычно

начало работы любой операционной системы сопровождается рассылкой широковещательных кадров, в которых станция заявляет о своем существовании и одновременно ищет серверы сети.

На этапе 1 станция посылает первый кадр с широковещательным адресом назначения и адресом источника 123 в свой сегмент. Кадр попадает как в мост 1, так и в мост 2. В обоих мостах новый адрес источника 123 заносится в адресную таблицу с пометкой о его принадлежности сегменту 1, то есть создается новая запись вида:

MAC-адрес 123 – Порт 1.

Так как адрес назначения широковещательный, то каждый мост должен передать кадр на сегмент 2. Эта передача происходит поочередно в соответствии с методом случайного доступа технологии Ethernet. Пусть первым доступ к сегменту 2 получает мост 1 (этап 2 на рис. 13.7). При появлении кадра на сегменте 2 мост 2 принимает его в свой буфер и обрабатывает. Он видит, что адрес 123 уже есть в его адресной таблице, но пришедший кадр является более свежим, и он решает, что адрес 123 принадлежит сегменту 2, а не 1. Поэтому мост 2 корректирует содержимое базы и делает запись о том, что адрес 123 принадлежит сегменту 2:

MAC-адрес 123 – Порт 2.

Аналогично поступает мост 1, когда мост 2 передает свою копию кадра на сегмент 2.

Далее перечислены последствия наличия петли в сети.

- «Размножение» кадра, то есть появление нескольких его копий (в данном случае – двух, но если бы сегменты были соединены тремя мостами – то трех и т. д.).
- Бесконечная циркуляция обеих копий кадра по петле в противоположных направлениях, а значит, засорение сети ненужным трафиком.
- Постоянная перестройка мостами своих адресных таблиц, так как кадр с адресом источника 123 будет появляться то на одном порту, то на другом.

В целях исключения всех этих нежелательных эффектов мосты/коммутаторы нужно применять так, чтобы между логическими сегментами не было петель, то есть строить с помощью коммутаторов только древовидные структуры, гарантирующие наличие единственного пути между любыми двумя сегментами. Тогда кадры от каждой станции будут поступать на мост/коммутатор всегда с одного и того же порта, и коммутатор сможет правильно решать задачу выбора рационального маршрута в сети.

В небольших сетях сравнительно легко гарантировать наличие одного и только одного пути между двумя сегментами. Но когда количество соединений возрастает, то вероятность непреднамеренного образования петли оказывается высокой.

Возможна и другая причина возникновения петель. Так, для повышения надежности желательно иметь между мостами/коммутаторами резервные связи, которые не участвуют в нормальной работе основных связей по передаче информационных кадров станций, но при отказе какой-либо основной связи образуют новую связную рабочую конфигурацию без петель.

Избыточные связи необходимо блокировать, то есть переводить их в неактивное состояние. В сетях с простой топологией эта задача решается вручную путем блокирования соответствующих портов мостов/коммутаторов. В больших сетях со сложными связями используются алгоритмы, которые позволяют решать задачу обнаружения петель автоматически. Наиболее известным из них является стандартный **алгоритм покрывающего дерева** (Spanning Tree Algorithm, STA), который будет детально рассмотрен в главе 14.

Коммутаторы

Параллельная коммутация

При появлении в конце 80-х начале 90-х годов быстрых протоколов, производительных персональных компьютеров, мультимедийной информации и разделении сети на большое количество сегментов классические *мосты* перестали справляться с работой. Обслуживание потоков кадров между теперь уже несколькими портами с помощью одного процессорного блока требовало значительного повышения быстродействия процессора, а это довольно дорогостоящее решение.

Более эффективным оказалось решение, которое и «породило» коммутаторы: для обслуживания потока, поступающего на каждый порт, в устройство ставился отдельный специализированный процессор, который реализовывал алгоритм прозрачного моста. По сути, коммутатор — это мультипроцессорный мост, способный параллельно продвигать кадры сразу между всеми парами своих портов. Но если при добавлении процессорных блоков компьютер не перестали называть компьютером, а добавили только прилагательное «мультипроцессорный», то с мультипроцессорными мостами произошла метаморфоза — во многом по маркетинговым причинам они превратились в коммутаторы. Нужно отметить, что помимо процессоров портов коммутатор имеет центральный процессор, который координирует работу портов, отвечая за построение общей таблицы продвижения, а также поддерживая функции конфигурирования и управления коммутатором.

Со временем коммутаторы вытеснили из локальных сетей классические однопроцессорные мосты. Основная причина этого — существенно более высокая производительность, с которой коммутаторы передают кадры между сегментами сети. Если мосты могли даже замедлять работу сети, то коммутаторы всегда выпускаются с процессорами портов, способными передавать кадры с той максимальной скоростью, на которую рассчитан протокол. Ну а добавление к этому возможности параллельной передачи кадров между портами предопределило судьбу и мостов, и коммутаторов.

> Производительность коммутаторов на несколько порядков выше, чем мостов — коммутаторы могут передавать до нескольких десятков, а иногда и сотен миллионов кадров в секунду, в то время как мосты обычно обрабатывали 3–5 тысяч кадров в секунду.

За время своего существования уже без конкурентов-мостов коммутаторы вобрали в себя многие дополнительные функции, родившиеся в результате естественного развития сетевых технологий. К этим функциям относятся, например, поддержка виртуальных сетей (VLAN), агрегирование линий связи, приоритезация трафика и т. п. Развитие технологии производства заказных микросхем также способствовало успеху коммутаторов, в результате процессоры портов сегодня обладают такой вычислительной мощностью, которая позволяет им быстро реализовывать весьма сложные алгоритмы обработки трафика, например выполнять его классификацию и профилирование.

Технология коммутации сегментов Ethernet была предложена небольшой компанией Kalpana в 1990 году в ответ на растущие потребности в повышении пропускной способности связей высокопроизводительных серверов с сегментами рабочих станций. У коммутатора компании Kalpana при свободном в момент приема кадра состоянии выходного порта задержка между получением первого байта кадра и появлением этого же байта на

выходе порта адреса назначения составляла всего 40 мкс, что было гораздо ниже задержки кадра при его передаче мостом.

Структурная схема коммутатора EtherSwitch, предложенного фирмой Kalpana, представлена на рис. 13.8.

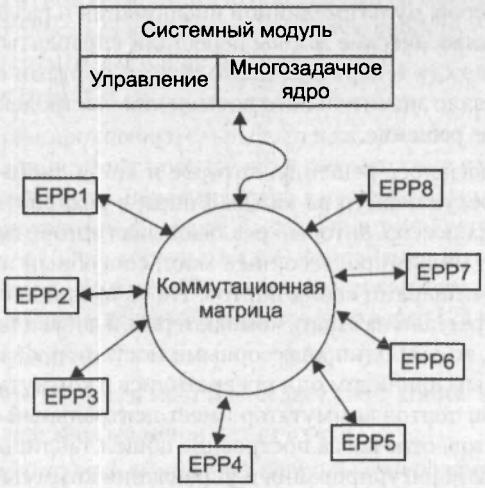


Рис. 13.8. Структура коммутатора EtherSwitch компании Kalpana

Каждый из 8 портов 10Base-T обслуживается одним **процессором пакетов Ethernet** (Ethernet Packet Processor, EPP). Кроме того, коммутатор имеет системный модуль, который координирует работу всех процессоров EPP, в частности ведет общую адресную таблицу коммутатора. Для передачи кадров между портами используется **коммутационная матрица**. Она функционирует по принципу коммутации каналов, соединяя порты коммутатора. Для 8 портов матрица может одновременно обеспечить 8 внутренних каналов при полудуплексном режиме работы портов и 16 — при дуплексном, когда передатчик и приемник каждого порта работают независимо друг от друга.

При поступлении кадра в какой-либо порт соответствующий процессор EPP буферизует несколько первых байтов кадра, чтобы прочитать адрес назначения. После получения адреса назначения процессор сразу же приступает к обработке кадра, не дожидаясь прихода остальных его байтов.

1. Процессор EPP просматривает свой кэш адресной таблицы, и если не находит там нужного адреса, обращается к системному модулю, который работает в многозадачном режиме, параллельно обслуживая запросы всех процессоров EPP. Системный модуль производит просмотр общей адресной таблицы и возвращает процессору найденную строку, которую тот буферизует в своем кэше для последующего использования.
2. Если адрес назначения найден в адресной таблице и кадр нужно отфильтровать, процессор просто прекращает записывать в буфер байты кадра, очищает буфер и ждет поступления нового кадра.
3. Если же адрес найден и кадр нужно передать на другой порт, процессор, продолжая прием кадра в буфер, обращается к коммутационной матрице, пытаясь установить в ней путь, связывающий его порт с портом, через который идет маршрут к адресу

назначения. Коммутационная матрица способна помочь только в том случае, если порт адреса назначения в этот момент свободен, то есть не соединен с другим портом данного коммутатора.

4. Если же порт занят, то, как и в любом устройстве с коммутацией каналов, матрица в соединении отказывает. В этом случае кадр полностью буферизуется процессором входного порта, после чего процессор ожидает освобождения выходного порта и образования коммутационной матрицей нужного пути.
5. После того как нужный путь установлен, в него направляются буферизованные байты кадра, которые принимаются процессором выходного порта. Как только процессор выходного порта получает доступ к подключенному к нему сегменту Ethernet по алгоритму CSMA/CD[1], байты кадра сразу же начинают передаваться в сеть. Процессор входного порта постоянно хранит несколько байтов принимаемого кадра в своем буфере, что позволяет ему независимо и асинхронно принимать и передавать байты кадра (рис. 13.9).

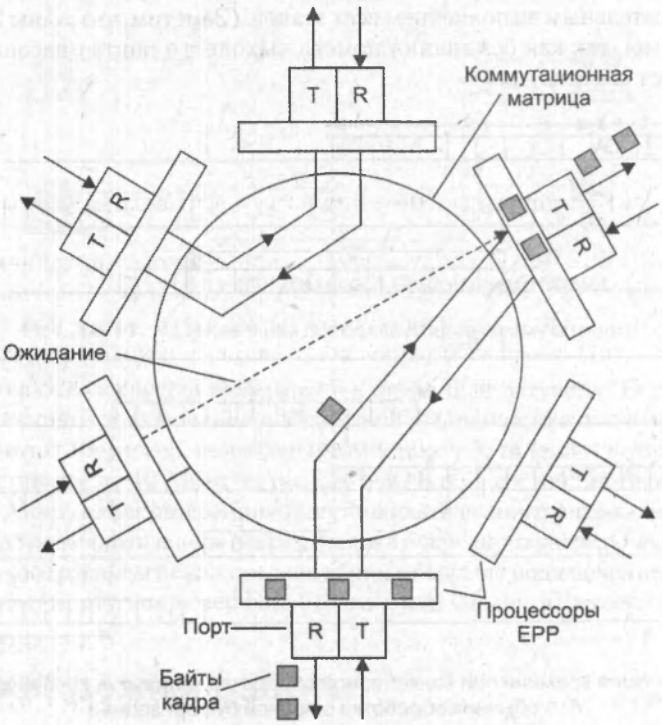


Рис. 13.9. Передача кадра через коммутационную матрицу

Описанный пособ передачи кадра без его полной буферизации получил название **коммутации «на лету»** (on-the-fly), или **«напролет»** (cut-through). Этот способ представляет

[1] Во время появления коммутатора Kalpana основным режимом работы сегментов был режим разделения среды.

собой, по сути, *конвейерную обработку* кадра, когда частично совмещаются во времени несколько этапов его передачи.

1. Прием первых байтов кадра процессором входного порта, включая прием байтов адреса назначения.
2. Поиск адреса назначения в адресной таблице коммутатора (в кэше процессора или в общей таблице системного модуля).
3. Коммутация матрицы.
4. Прием остальных байтов кадра процессором входного порта.
5. Прием байтов кадра (включая первые) процессором выходного порта через коммутационную матрицу.
6. Получение доступа к среде процессором выходного порта.
7. Передача байтов кадра процессором выходного порта в сеть.

На рис. 13.10 подставлены два режима обработки кадра: режим коммутации «на лету» с частичным совмещением во времени нескольких этапов и режим полной буферизации кадра с последовательным выполнением всех этапов. (Заметим, что этапы 2 и 3 совместить во времени нельзя, так как без знания номера выходного порта операция коммутации матрицы не имеет смысла.)

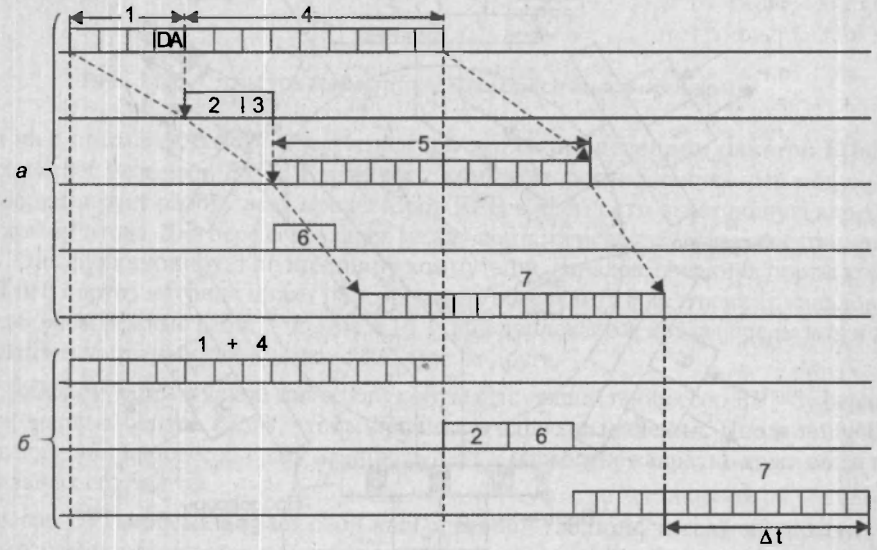


Рис. 13.10. Экономия времени при конвейерной обработке кадра: *а* — конвейерная обработка, *б* — обычная обработка с полной буферизацией

Как показывает схема, экономия от конвейеризации получается ощутимой.

Однако главной причиной повышения производительности сети при использовании коммутатора является *параллельная* обработка нескольких кадров.

Этот эффект иллюстрирует рис. 13.11, на котором показана идеальная в отношении производительности ситуация, когда четыре порта из восьми передают данные с максимальной для протокола Ethernet скоростью в 10 Мбит/с. Причем они передают эти данные на

остальные четыре порта коммутатора не конфликтуя: потоки данных между узлами сети распределились так, что для каждого принимающего кадры порта есть свой выходной порт. Если коммутатор успевает обрабатывать входной трафик при максимальной интенсивности поступления кадров на входные порты, то общая производительность коммутатора в приведенном примере составит 4 × 10 = 40 Мбит/с, а при обобщении примера для N портов — (N/2) × 10 Мбит/с. В таком случае говорят, что *коммутатор предоставляет каждой станции или сегменту, подключенному к его портам, выделенную пропускную способность протокола.*

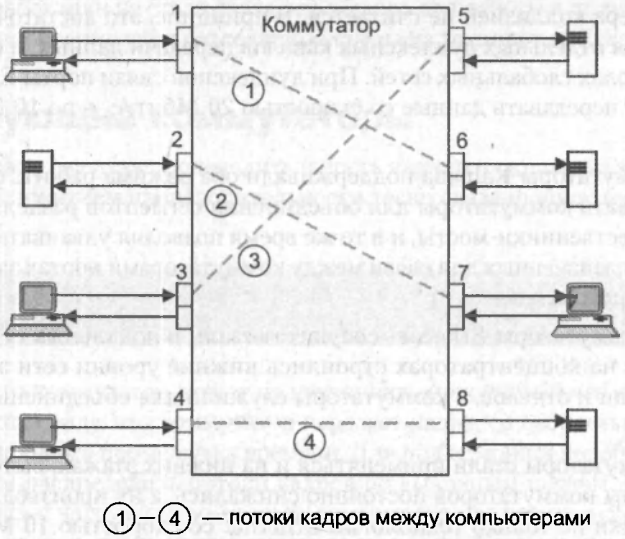


Рис. 13.11. Параллельная передача кадров коммутатором

Естественно, что в сети не всегда складывается описанная ситуация. Если двум станциям, например станциям, подключенным к портам 3 и 4, одновременно нужно записывать данные на один и тот же сервер, подключенный к порту 8, то коммутатор не сможет выделить каждой станции по 10 Мбит/с, так как порт 8 не в состоянии передавать данные со скоростью 20 Мбит/с. Кадры станций будут ожидать во внутренних очередях входных портов 3 и 4, когда освободится порт 8 для передачи очередного кадра. Очевидно, хорошим решением для такого распределения потоков данных было бы подключение сервера к более высокоскоростному порту, например Fast Ethernet или Gigabit Ethernet.

Дуплексный режим работы

Технология коммутации сама по себе не имеет непосредственного отношения к методу доступа к среде, который используется портами коммутатора. При подключении к порту коммутатора сегмента, иредставляющего собой разделяемую среду, данный порт, как и все остальные узлы такого сегмента, должен поддерживать полудуплексный режим.

Однако когда к каждому порту коммутатора подключен не сегмент, а только *один* компьютер, причем по двум физически раздельным каналам, как это происходит почти во всех стандартах Ethernet, кроме коаксиальных версий Ethernet, ситуация становится не такой однозначной. Порт может работать как в обычном полудуплексном режиме, так и в дуплексном.

В **полудуплексном режиме** работы порт коммутатора по-прежнему распознает коллизии. Доменом коллизий в этом случае является участок сети, включающий передатчик коммутатора, приемник коммутатора, передатчик сетевого адаптера компьютера, приемник сетевого адаптера компьютера и две витые пары, соединяющие передатчики с приемниками.

Коллизия возникает, когда передатчики порта коммутатора и сетевого адаптера одновременно или почти одновременно начинают передачу своих кадров.

В **дуплексном режиме** одновременная передача данных передатчиком порта коммутатора и сетевого адаптера коллизией не считается. В принципе, это достаточно естественный режим работы для отдельных дуплексных каналов передачи данных, и он всегда использовался в протоколах глобальных сетей. При дуплексной связи порты Ethernet стандарта 10 Мбит/с могут передавать данные со скоростью 20 Мбит/с — по 10 Мбит/с в каждом направлении.

Уже первые коммутаторы Kalpana поддерживали оба режима работы своих портов, позволяя использовать коммутаторы для объединения сегментов разделяемой среды, как делали их предшественники-мосты, и в то же время позволяя удваивать скорость обмена данными на предназначенных для связи между коммутаторами портах за счет работы этих портов в дуплексном режиме.

Долгое время коммутаторы Ethernet сосуществовали в локальных сетях с концентраторами Ethernet: на концентраторах строились нижние уровни сети здания, такие как сети рабочих групп и отделов, а коммутаторы служили для объединения этих сегментов в общую сеть.

Постепенно коммутаторы стали применяться и на нижних этажах, вытесняя концентраторы, так как цены коммутаторов постоянно снижались, а их производительность росла (за счет поддержки не только технологии Ethernet со скоростью 10 Мбит/с, но и всех последующих более скоростных версий этой технологии, то есть Fast Ethernet со скоростью 100 Мбит/с, Gigabit Ethernet со скоростью 1 Гбит/с и 10G Ethernet со скоростью 10 Гбит/с). Этот процесс завершился вытеснением концентраторов Ethernet и переходом к полностью коммутируемым сетям, пример такой сети показан на рис. 13.12.

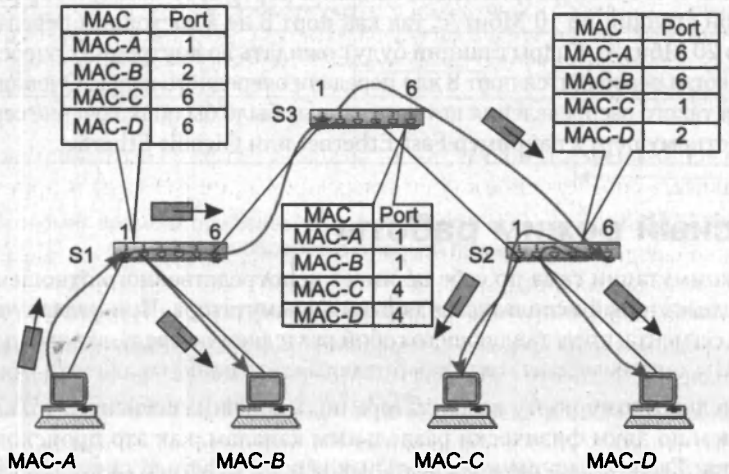


Рис. 13.12. Полностью коммутируемая сеть Ethernet

В полностью коммутируемой сети Ethernet все порты работают в дуплексном режиме, а продвижение кадров осуществляется на основе MAC-адресов.

При разработке технологий Fast Ethernet и Gigabit Ethernet дуплексный режим стал одним из двух полноправных стандартных режимов работы узлов сети. Однако уже практика применения первых коммутаторов с портами Gigabit Ethernet показала, что они практически всегда применяются в дуплексном режиме для взаимодействия с другими коммутаторами или высокоскоростными сетевыми адаптерами. Поэтому при разработке стандарта 10G Ethernet его разработчики не стали создавать версию для работы в полудуплексном режиме, окончательно закрепив уход разделяемой среды из технологии Ethernet.

Неблокирующие коммутаторы

Как уже отмечалось, высокая производительность является одним из главных достоинств коммутаторов. С понятием производительности тесно связано понятие неблокирующего коммутатора.

Коммутатор называют **неблокирующим**, если он может передавать кадры через свои порты с той же скоростью, с которой они на них поступают.

Когда говорят, что коммутатор может поддерживать *устойчивый неблокирующий режим работы*, то имеют в виду, что коммутатор передает кадры со скоростью их поступления в течение произвольного промежутка времени. Для поддержания подобного режима нужно таким образом распределить потоки кадров по выходным портам, чтобы, во-первых, порты справлялись с нагрузкой, во-вторых, коммутатор мог всегда в среднем передать на выходы столько кадров, сколько их поступило на входы. Если же входной поток кадров (просуммированный по всем портам) в среднем будет превышать выходной поток кадров (также просуммированный по всем портам), то кадры будут накапливаться в буферной памяти коммутатора и при переполнении просто отбрасываться.

Для поддержания устойчивого неблокирующего режима работы коммутатора необходимо, чтобы его производительность удовлетворяла условию $C_k = (\Sigma C_{pi})/2$, где C_k — производительность коммутатора, C_{pi} — максимальная производительность протокола, поддерживаемого i-м портом коммутатора.

В этом соотношении под производительностью коммутатора в целом понимается его способность продвигать определенное количество кадров, принимаемых от приемников всех его портов, на передатчики всех его портов.

В суммарной производительности портов каждый проходящий кадр учитывается дважды, как входящий и как выходящий, а так как в устойчивом режиме входной трафик равен выходному, то минимально достаточная производительность коммутатора для поддержки неблокирующего режима равна половине суммарной производительности портов. Если порт, например, стандарта Ethernet со скоростью 10 Мбит/с работает в полудуплексном режиме, то производительность порта C_{pi} равна 10 Мбит/с, а если в дуплексном — 20 Мбит/с.

Иногда говорят, что коммутатор поддерживает *мгновенный неблокирующий режим*. Это означает, что он может принимать и обрабатывать кадры от всех своих портов на максимальной скорости протокола независимо от того, обеспечиваются ли условия устойчивого

равновесия между входным и выходным трафиком. Правда, обработка некоторых кадров при этом может быть неполной — при занятости выходного порта кадр помещается в буфер коммутатора.

Для поддержки мгновенного неблокирующего режима коммутатор должен обладать большей собственной производительностью, а именно она должна быть равна суммарной производительности его портов: $C_k = \Sigma C_{pi}$.

Приведенные соотношения справедливы для портов с любыми скоростями, то есть портов стандартов Ethernet со скоростью 10 Мбит/с, Fast Ethernet, Gigabit Ethernet и 10G Ethernet.

Способы, которыми обеспечивается способность коммутатора поддерживать неблокирующий режим, могут быть разными. Необходимым требованием является умение процессора порта обрабатывать потоки кадров с максимальной для физического уровня этого порта скоростью. В главе 12 мы подсчитали, что максимальная производительность порта Ethernet стандарта 10 Мбит/с равна 14 880 кадров в секунду. Это означает, что процессоры портов Ethernet стандарта 10 Мбит/с неблокирующего коммутатора должны поддерживать продвижение кадров со скоростью 14 880 кадров в секунду.

Однако только адекватной производительности процессоров портов недостаточно для того, чтобы коммутатор был неблокирующим. Необходимо, чтобы достаточной производительностью обладали все элементы архитектуры коммутатора, включая центральный процессор, общую память, шины, соединяющие отдельные модули между собой, саму архитектуру коммутатора (наиболее распространенные архитектуры коммутаторов мы рассмотрим позже). В принципе, задача создания неблокирующего коммутатора аналогична задаче создания высокопроизводительного компьютера — в обоих случаях она решается комплексно: за счет соответствующей архитектуры объединения модулей в едином устройстве и адекватной производительности каждого отдельного модуля устройства.

Борьба с перегрузками

Даже в том случае, когда коммутатор является неблокирующим, нет гарантии того, что он во всех случаях справится с потоком кадров, направляемых на его порты. Неблокирующие коммутаторы тоже могут испытывать перегрузки и терять кадры из-за переполнения внутренних буферов.

Причина перегрузок обычно кроется не в том, что коммутатору не хватает производительности для обслуживания потоков кадров, а в ограниченной пропускной способности отдельного выходного порта, которая определяется параметрами протокола. Другим словами, какой бы производительностью коммутатор не обладал, всегда найдется такое распределение потоков кадров, которое приведет к перегрузке коммутатора из-за ограниченной производительности выходного порта коммутатора.

Возникновение таких перегрузок является платой за отказ от применения алгоритма доступа к разделяемой среде, так как в дуплексном режиме работы портов теряется контроль за потоками кадров, направляемых конечными узлами в сеть. В полудуплексном режиме, свойственном технологиям с разделяемой средой, поток кадров регулировался самим методом доступа к разделяемой среде. При переходе на дуплексный режим узлу разрешается отправлять кадры в коммутатор всегда, когда это ему нужно, поэтому в данном режиме коммутаторы сети могут сталкиваться с перегрузками, не имея при этом никаких средств «притормаживания» потока кадров.

Таким образом, если входной трафик неравномерно распределяется между выходными портами, легко представить ситуацию, когда на какой-либо выходной порт коммутатора будет направляться трафик с суммарной средней интенсивностью большей, чем протокольный максимум. На рис. 13.13 показана как раз такая ситуация, когда на порт 3 коммутатора Ethernet направляется от портов 1, 2, 4 и 6 поток кадров размером в 64 байт с суммарной интенсивностью в 22 100 кадров в секунду. Вспомним, что максимальная скорость в кадрах в секунду для сегмента Ethernet составляет 14 880. Естественно, что когда кадры поступают в буфер порта со скоростью 22 100 кадров в секунду, а уходят со скоростью 14 880 кадров в секунду, то внутренний буфер выходного порта начинает неуклонно заполняться необработанными кадрами.

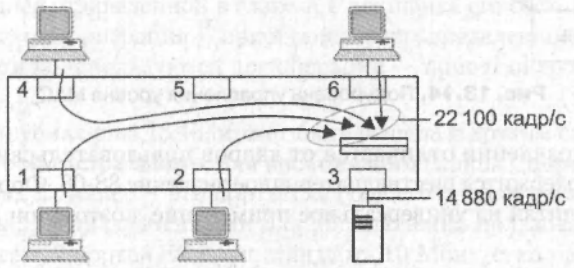


Рис. 13.13. Переполнение буфера порта из-за несбалансированности трафика

В приведенном примере нетрудно подсчитать, что при размере буфера в 100 Кбайт полное заполнение буфера произойдет через 0,22 секунды после начала работы в таком интенсивном режиме. Увеличение размера буфера до 1 Мбайт даст увеличение времени заполнения буфера до 2,2 секунды, что также неприемлемо. Проблему можно решить с помощью *средств контроля перегрузки*, которые были рассмотрены в главе 7.

Как мы знаем, существуют различные средства контроля перегрузки: управление очередями в коммутаторах, обратная связь, резервирование пропускной способности. На основе этих средств можно создать эффективную систему поддержки показателей QoS для трафика разных классов.

В этом разделе мы рассмотрим **механизм обратной связи**, который был стандартизован для сетей Ethernet в марте 1997 как спецификация IEEE 802.3x. Механизм обратной связи 802.3x используется только в дуплексном режиме работы портов коммутатора. Этот механизм очень важен для коммутаторов локальных сетей, так как он позволяет сократить потери кадров из-за переполнения буферов независимо от того, обеспечивает сеть дифференцированную поддержку показателей QoS для разных типов трафика или же предоставляет базовый сервис по доставке с максимальными усилиями («по возможности»). Другие механизмы поддержания показателей QoS рассматриваются в следующей главе.

Спецификация 802.3x вводит новый подуровень в стеке протоколов Ethernet — **подуровень управления уровня MAC**. Он располагается над уровнем MAC и является необязательным (рис. 13.14).

Кадры этого подуровня могут использоваться в различных целях, но пока в стандартах Ethernet для них определена только одна задача — приостановка передачи кадров другими узлами на определенное время.

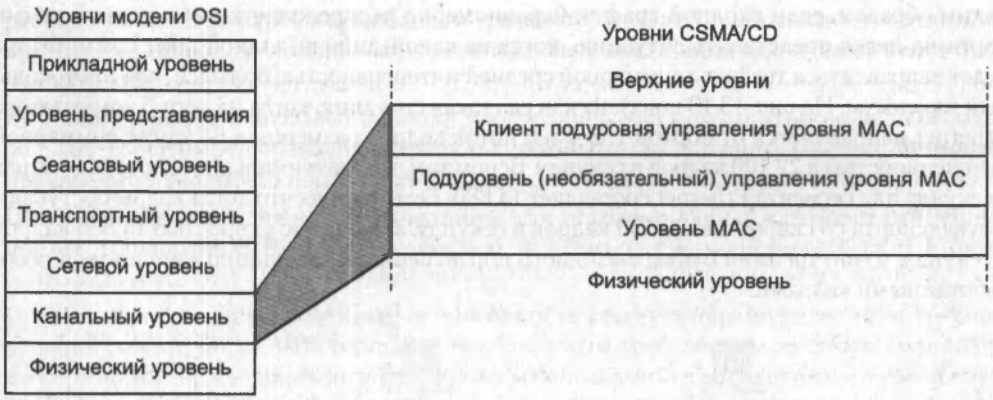


Рис. 13.14. Подуровень управления уровня MAC

Кадр подуровня управления отличается от кадров пользовательских данных тем, что в поле типа всегда содержится шестнадцатеричное значение 88-08. Формат кадра подуровня управления рассчитан на универсальное применение, поэтому он достаточно сложен (рис. 13.15).

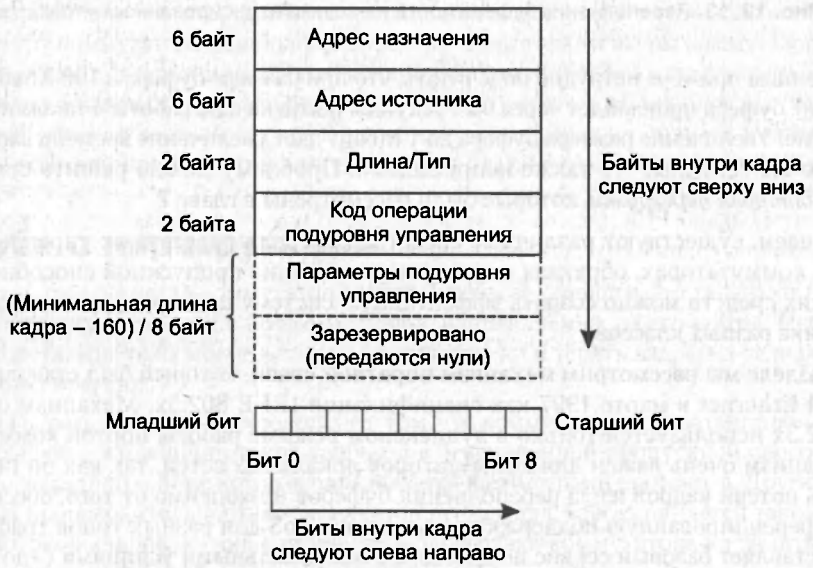


Рис. 13.15. Формат кадра подуровня управления

Коммутатор использует кадр подуровня управления в том случае, когда ему нужно на время приостановить поступление кадров от соседнего узла, чтобы разгрузить свои внутренние очереди.

В качестве адреса назначения можно указывать зарезервированное для этой цели значение группового адреса 01-80-C2-00-00-01. Это удобно, когда соседний узел также является

коммутатором (так как порты коммутатора не имеют уникальных MAC-адресов). Если сосед — конечный узел, можно также использовать уникальный MAC-адрес.

В поле кода операции подуровня управления указывается шестнадцатеричный код 00-01, поскольку, как уже было отмечено, пока определена только одна операция подуровня управления — она называется *PAUSE* (пауза) и имеет шестнадцатеричный код 00-01.

В поле параметров подуровня управления указывается время, на которое узел, получивший такой код, должен прекратить передачу кадров узлу, отправившему кадр с операцией *PAUSE*. Время измеряется в 512 битовых интервалах конкретной реализации Ethernet, диапазон возможных вариантов приостановки равен 0–65535.

Как видно из описания, этот механизм обратной связи относится к типу 2 в соответствии с классификацией, приведенной в главе 7. Специфика его состоит в том, что в нем предусмотрена только одна операция — приостановка на определенное время. Обычно же в механизмах этого типа используются две операции — приостановка и возобновление передачи кадров.

Проблема, иллюстрируемая рис. 13.13, может быть решена и другим способом: применением так называемого **магистрального**, или **восходящего** (uplink), **порта**. Магистральные порты в коммутаторах Ethernet — это порты следующего уровня иерархии скорости по сравнению с портами, предназначенными для подключения пользователей. Например, если коммутатор имеет 12 портов Ethernet стандарта 10 Мбит/с, то магистральный порт должен быть портом Fast Ethernet, чтобы его скорость была достаточна для передачи до 10 потоков от входных портов. Обычно низкоскоростные порты коммутатора служат для соединения с пользовательскими компьютерами, а магистральные порты — для подключения либо сервера, к которому обращаются пользователи, либо коммутатора более высокого уровня иерархии.

На рис. 13.16 показан пример коммутатора, имеющего 24 порта стандарта Fast Ethernet со скоростью 100 Мбит/с, к которым подключены пользовательские компьютеры, и один порт стандарта Gigabit Ethernet со скоростью 1000 Мбит/с, к которому подключен сервер. При такой конфигурации коммутатора вероятность перегрузки портов существенно снижается по сравнению с вариантом, когда все порты поддерживают одинаковую скорость. Хотя возможность перегрузки по-прежнему существует, для этого необходимо, чтобы более чем 10 пользователей одновременно обменивались с сервером данными со средней скоростью, близкой к максимальной скорости их соединений — а такое событие достаточно маловероятно.

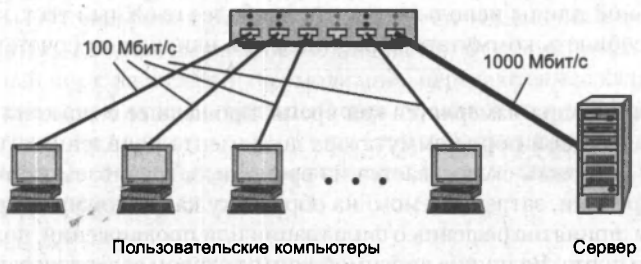


Рис. 13.16. Коммутатор рабочей группы

Из приведенного примера видно, что вероятность перегрузки портов коммутаторов зависит от распределения трафика между его портами, кроме того, понятно, что даже при хорошем

соответствии скорости портов наиболее вероятному распределению трафика полностью исключить перегрузки невозможно.

Поэтому в общем случае для уменьшения потерь кадров из-за перегрузок нужно применять оба средства: подбор скорости портов для наиболее вероятного распределения трафика в сети и протокол 802.3x для снижения скорости источника трафика в тех случаях, когда перегрузки все-таки возникают.

Характеристики производительности коммутаторов

Скорости фильтрации и продвижения кадров — две основные характеристики производительности коммутатора. Эти характеристики являются интегральными, они не зависят от того, каким образом технически реализован коммутатор.

Скорость фильтрации — это скорость, с которой коммутатор выполняет следующие этапы обработки кадров:

1. Прием кадра в свой буфер.
2. Просмотр адресной таблицы с целью нахождения порта для адреса назначения кадра.
3. Уничтожение кадра, так как его порт назначения и порт источника принадлежат одному логическому сегменту.

Скорость фильтрации практически у всех коммутаторов блокирующим фактором не является — коммутатор успевает отбрасывать кадры в темпе их поступления.

Скорость продвижения — это скорость, с которой коммутатор выполняет следующие этапы обработки кадров.

1. Прием кадра в свой буфер.
2. Просмотр адресной таблицы с целью нахождения порта для адреса назначения кадра.
3. Передача кадра в сеть через найденный по адресной таблице порт назначения.

Как скорость фильтрации, так и скорость продвижения измеряются обычно в кадрах в секунду. Если в характеристиках коммутатора не уточняется, для какого протокола и для какого размера кадра приведены значения скоростей фильтрации и продвижения, то по умолчанию считается, что эти показатели даются для протокола Ethernet и кадров минимального размера, то есть кадров длиной 64 байт. Как мы уже обсуждали, режим передачи кадров минимальной длины используется как наиболее сложный тест, который должен подтвердить способность коммутатора работать при наихудшем сочетании параметров трафика.

Задержка передачи кадра измеряется как время, прошедшее с момента прихода первого байта кадра на входной порт коммутатора до момента появления этого байта на его выходном порту. Задержка складывается из времени, затрачиваемого на буферизацию байтов кадра, и времени, затрачиваемого на обработку кадра коммутатором — просмотр адресной таблицы, принятие решения о фильтрации или продвижении, получение доступа к среде выходного порта. Величина вносимой коммутатором задержки зависит от режима его работы. Если коммутация осуществляется «на лету», то задержки обычно невелики и составляют от 5 до 40 мкс, а при полной буферизации кадров — от 50 до 200 мкс для кадров минимальной длины при передаче со скоростью 10 Мбит/с. Коммутаторы, под-

держивающие более скоростные версии Ethernet, вносят меньшие задержки в процесс продвижения кадров.

Производительность коммутатора определяется количеством пользовательских данных, переданных в единицу времени через его порты, и измеряется в мегабитах в секунду (Мбит/с). Так как коммутатор работает на канальном уровне, для него пользовательскими данными являются те данные, которые переносятся в поле данных кадров Ethernet. Максимальное значение производительности коммутатора всегда достигается на кадрах максимальной длины, так как при этом доля накладных расходов на служебную информацию кадра минимальна. Коммутатор — это многопортовое устройство, поэтому для него в качестве характеристики принято давать максимальную суммарную производительность при одновременной передаче трафика по всем его портам.

Еще одной важной конструктивной характеристикой коммутатора является **максимальная емкость адресной таблицы**. Она определяет предельное количество MAC-адресов, которыми может одновременно оперировать коммутатор.

Для выполнения операций каждого порта в коммутаторах чаще всего используется выделенный процессорный блок со своей памятью для хранения собственного экземпляра адресной таблицы. Каждый порт хранит только те наборы адресов, с которыми он работал в последнее время, поэтому экземпляры адресной таблицы разных процессорных модулей, как правило, не совпадают.

Значение максимального числа MAC-адресов, которое может запомнить процессор порта, зависит от области применения коммутатора. Коммутаторы рабочих групп обычно поддерживают всего несколько адресов на порт, так как они предназначены для образования микросегментов. Коммутаторы отделов должны поддерживать несколько сотен адресов, а коммутаторы магистралей сетей — до нескольких тысяч (обычно 4000–8000 адресов).

Недостаточная емкость адресной таблицы может служить причиной замедления работы коммутатора и засорения сети избыточным трафиком. Если адресная таблица процессора порта полностью заполнена, а он встречает новый адрес источника в поступившем кадре, процессор должен удалить из таблицы какой-либо старый адрес и поместить на его место новый. Эта операция сама по себе отнимает у процессора часть времени, но главные потери производительности наблюдаются при поступлении кадра с адресом назначения, который пришлось удалить из адресной таблицы. Так как адрес назначения кадра неизвестен, коммутатору приходится передавать этот кадр на все остальные порты. Некоторые производители коммутаторов решают эту проблему за счет изменения алгоритма обработки кадров с неизвестным адресом назначения. Один из портов коммутатора конфигурируется как магистральный порт, на который по умолчанию передаются все кадры с неизвестным адресом[1]. Передача кадра на магистральный порт производится в расчете на то, что этот порт подключен к вышестоящему коммутатору (при иерархическом соединении коммутаторов в крупной сети), который имеет достаточную емкость адресной таблицы и «знает», куда можно передать любой кадр.

[1] В маршрутизаторах такой прием применяется давно, позволяя сократить размеры адресных таблиц в сетях, организованных по иерархическому принципу.

Скоростные версии Ethernet

Скорость 10 Мбит/с первой стандартной версии Ethernet долгое время удовлетворяла потребности пользователей локальных сетей. Однако в начале 90-х годов начала ощущаться недостаточная пропускная способность Ethernet, так как скорость обмена с сетью стала существенно меньше скорости внутренней шины компьютера. Кроме того, начали появляться новые мультимедийные приложения, гораздо более требовательные к скорости сети, чем их текстовые предшественники. В поисках решения проблемы ведущие производители сетевого оборудования начали интенсивные работы по повышению скорости Ethernet при сохранении главного достоинства этой технологии — простоты и низкой стоимости оборудования.

Результатом стало появление новых скоростных стандартов Ethernet: Fast Ethernet (скорость 100 Мбит/с), Gigabit Ethernet (1000 Мбит/с, или 1 Гбит/с) и 10G Ethernet (10 Гбит/с). На время написания этой книги два новых стандарта — 40G Ethernet и 100G Ethernet — находились в стадии разработки, обещая следующее десятикратное превышение верхней границы производительности Ethernet.

Разработчикам новых скоростных стандартов Ethernet удалось сохранить основные черты классической технологии Ethernet и, прежде всего, простой способ обмена кадрами без встроенных в технологию сложных контрольных процедур. Этот фактор оказался решающим в соревновании технологий локальных сетей, так как выбор пользователей всегда склонялся в пользу простого наращивания скорости сети, а не в пользу решений, связанных с более эффективным расходованием той же самой пропускной способности с помощью более сложной и дорогой технологии. Примером такого подхода служит переход с оборудования Fast Ethernet на Gigabit Ethernet вместо перехода на оборудование ATM со скоростью 155 Мбит/с. Несмотря на значительную разницу в пропускной способности (1000 Мбит/с против 155 Мбит/с), оба варианта обновления сети примерно равны по степени положительного влияния на «самочувствие» приложений, так как Gigabit Ethernet достигает нужного эффекта за счет равного повышения доли пропускной способности для всех приложений, а ATM перераспределяет меньшую пропускную способность более тонко, дифференцируя ее в соответствии с потребностями приложений. Тем не менее пользователи предпочли не вдаваться в детали и тонкости настройки сложного оборудования, когда можно просто применить знакомое и простое, но более скоростное оборудование Ethernet.

Значительный вклад в «победу» Ethernet внесли также коммутаторы локальных сетей, так как их успех привел к отказу от разделяемой среды, где технология Ethernet всегда была уязвимой из-за случайного характера метода доступа. Начиная с версии 10G Ethernet, разработчики перестали включать вариант работы на разделяемой среде в описание стандарта. Коммутаторы с портами Fast Ethernet, Gigabit Ethernet и 10G Etherhet работают по одному и тому же алгоритму, описанному в стандарте IEEE 802.1D. Возможность комбинировать порты с различными скоростями в диапазоне от 10 Мбит/с до 10 Гбит/с делает коммутаторы Ethernet гибкими и эффективными сетевыми устройствами, позволяющими строить разнообразные сети.

Повышение скорости работы Ethernet было достигнуто за счет улучшения качества кабелей, применяемых в компьютерных сетях, а также совершенствования методов кодирования данных при их передаче по кабелям, то есть за счет совершенствования физического уровня технологии.

Fast Ethernet

История создания

В 1992 году группа производителей сетевого оборудования, включая таких лидеров технологии Ethernet, как SynOptics, 3Com и ряд других, образовала некоммерческое объединение *Fast Ethernet Alliance* для разработки стандарта новой технологии, которая должна была обеспечить резкое повышение производительности при максимально возможном сохранении особенностей технологии Ethernet.

В комитете 802 института IEEE в это же время была сформирована исследовательская группа для изучения технического потенциала новых высокоскоростных технологий. За период с конца 1992 года и по конец 1993 года группа IEEE изучила 100-мегабитные решения, предложенные различными производителями. Наряду с предложениями Fast Ethernet Alliance группа рассмотрела также и высокоскоростную технологию, предложенную компаниями Hewlett-Packard и AT&T.

В центре дискуссий была проблема сохранения метода случайного доступа CSMA/CD. Предложение Fast Ethernet Alliance сохраняло этот метод и тем самым обеспечивало преемственность и согласованность сетей со скоростями 10 Мбит/с и 100 Мбит/с. Коалиция HP и AT&T, которая заручилась поддержкой значительно меньшего числа производителей в сетевой индустрии, чем Fast Ethernet Alliance, предложила совершенно новый метод доступа, названный **приоритетным доступом по требованию** (demand priority). Он существенно менял картину поведения узлов в сети, поэтому не смог вписаться в технологию Ethernet и стандарт 802.3; для его стандартизации был организован новый комитет IEEE 802.12.

Осенью 1995 года обе технологии стали стандартами IEEE. Комитет IEEE 802.3 принял спецификацию Fast Ethernet в качестве стандарта 802.3u, который не является самостоятельным стандартом, а представляет собой дополнение к существующему стандарту 802.3 в виде глав с 21 по 30. Комитет 802.12 принял технологию 100VG-AnyLAN, в которой использовался приоритетный доступ по требованию и поддерживались кадры двух форматов – Ethernet и Token Ring.

Технологии Fast Ethernet и 100VG-AnyLAN в первые месяцы своего существования рассматривались как равные соперники, но очень скоро стало ясно, что пользователи предпочитают более простую и знакомую технологию Fast Ethernet. Вскоре технология 100VG-AnyLAN прекратила свое существование; немаловажным фактором этого стал и переход локальных сетей на полностью коммутируемые версии, сводящий «на нет» преимущества более совершенного метода доступа технологии 100VG-AnyLAN.

Физические уровни технологии Fast Ethernet

Все отличия технологий Fast Ethernet и Ethernet сосредоточены на физическом уровне (рис. 13.17). Уровни MAC и LLC в Fast Ethernet остались абсолютно теми же, и их описывают прежние главы стандартов 802.3 и 802.2. Поэтому, рассматривая технологию Fast Ethernet, мы будем изучать только несколько вариантов ее физического уровня.

Организация физического уровня технологии Fast Ethernet является более сложной, поскольку в ней используются *три* варианта кабельных систем:

- волоконно-оптический многомодовый кабель (два волокна);
- витая пара категории 5 (две пары);
- витая пара категории 3 (четыре пары).

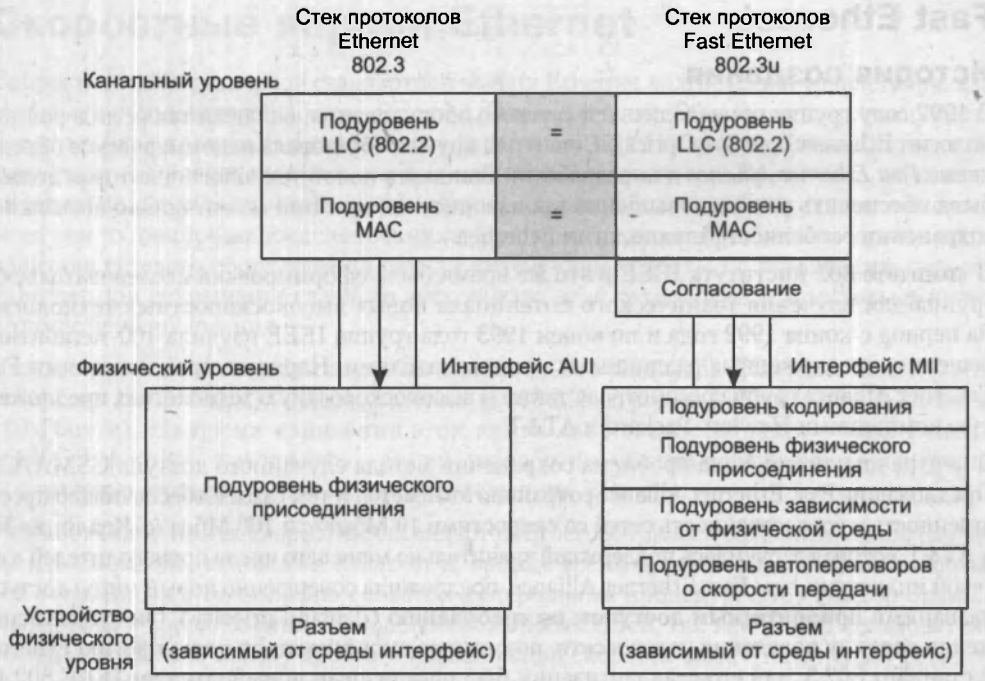


Рис. 13.17. Отличия технологий Fast Ethernet и Ethernet

Коаксиальный кабель, давший миру первую сеть Ethernet, в число разрешенных сред передачи данных новой технологии Fast Ethernet не попал. Это общая тенденция многих новых технологий, поскольку на небольших расстояниях витая пара категории 5 позволяет передавать данные с той же скоростью, что и коаксиальный кабель, но сеть получается более дешевой и удобной в эксплуатации. На больших расстояниях оптическое волокно обладает гораздо более широкой полосой пропускания, чем коаксиал, а стоимость сети получается ненамного выше, особенно если учесть высокие затраты на поиск и устранение неисправностей в крупной кабельной коаксиальной системе.

Официальный стандарт 802.3 установил три различных спецификации для физического уровня Fast Ethernet и дал им следующие названия (рис. 13.18):

- **100Base-TX** для двухпарного кабеля на неэкранированной витой паре UTP категории 5 или экранированной витой паре STP типа 1;
- **100Base-T4** для четырехпарного кабеля на неэкранированной витой паре UTP категории 3, 4 или 5;
- **100Base-FX** для многомодового оптоволоконного кабеля с двумя волокнами.

Для всех трех стандартов справедливы перечисленные далее утверждения и характеристики.

Форматы кадров технологии Fast Ethernet не отличаются от форматов кадров технологий 10-мегабитной сети Ethernet.

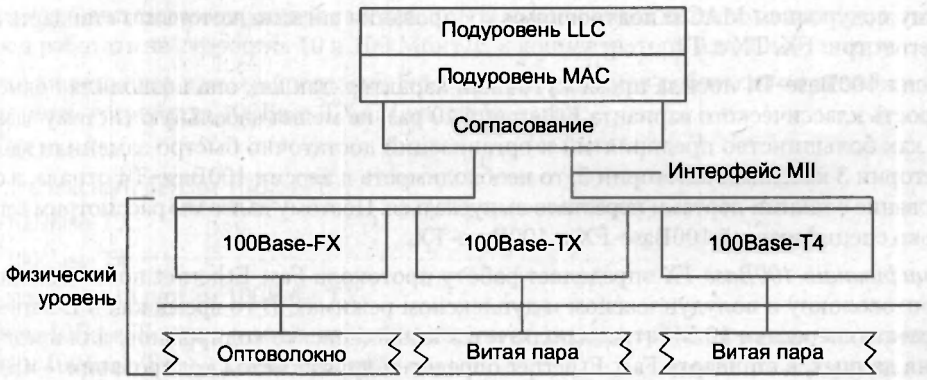


Рис. 13.18. Структура физического уровня Fast Ethernet

Межкадровый интервал равен 0,96 мкс, а битовый интервал — 10 нс. Все временные параметры алгоритма доступа (интервал отсрочки, время передачи кадра минимальной длины и т. п.), измеренные в битовых интервалах, остались прежними.

Признаком свободного состояния среды является передача по ней символа простоя источника — соответствующего избыточного кода (а не отсутствие сигналов, как в стандартах Ethernet со скоростью 10 Мбит/с).

Физический уровень включает три элемента.

- **Независимый от среды интерфейс** (Media Independent Interface, MII).
- **Уровень согласования** нужен для того, чтобы уровень MAC, рассчитанный на интерфейс AUI, мог работать с физическим уровнем через интерфейс MII.
- **Устройство физического уровня** (Physical Layer Device, PHY) состоит, в свою очередь, из нескольких подуровней (см. рис. 13.17):
  - подуровня логического кодирования данных, преобразующего поступающие от уровня MAC байты в символы кода 4B/5B или 8B/6T (первый метод кодирования используются в версиях 100Base-TX и 100Baase-FX, второй — в версии 100Base-T4);
  - подуровней физического присоединения и зависимости от физической среды (PMD), которые обеспечивают формирование сигналов в соответствии с методом физического кодирования, например NRZI или MLT-3;
  - подуровня автопереговоров, который позволяет двум взаимодействующим портам автоматически выбрать наиболее эффективный режим работы, например полудуплексный или дуплексный (этот подуровень является факультативным).

Интерфейс MII поддерживает независимый от физической среды способ обмена данными между подуровнем MAC и подуровнем PHY. Этот интерфейс аналогичен по назначению интерфейсу AUI классического стандарта Ethernet за исключением того, что интерфейс AUI располагался между подуровнем физического кодирования сигнала (для любых вариантов кабеля использовался одинаковый метод физического кодирования — манчестерский код) и подуровнем физического присоединения к среде, а интерфейс MII располагается

между подуровнем MAC и подуровнями кодирования сигнала, которых в стандарте Fast Ethernet три: FX, TX и T4.

Версия 100Base-T4 носила промежуточный характер, так как она позволяла повысить скорость классического варианта Ethernet в 10 раз, не меняя кабельную систему здания. Так как большинство предприятий и организаций достаточно быстро заменили кабели категории 3 кабелями категории 5, то необходимость в версии 100Base-T4 отпала, и оборудование с такими портами перестало выпускаться. Поэтому далее мы рассмотрим детали только спецификаций 100Base-FX и 100Base-TX.

Спецификация 100Base-FX определяет работу протокола Fast Ethernet по многомодовому оптоволокну в полудуплексном и дуплексном режимах. В то время как в Ethernet со скоростью передачи 10 Мбит/с используется манчестерское кодирование для представления данных, в стандарте Fast Ethernet определен другой метод кодирования — 4B/5B, который мы рассматривали в главе 9. Этот метод к моменту разработки технологии Fast Ethernet уже показал свою эффективность в сетях FDDI, поэтому он без изменений был перенесен в спецификацию 100Base-FX/TX. Напомним, что в этом методе каждые четыре бита данных подуровня MAC (называемых символами) представляются пятью битами. Избыточный бит позволяет применить потенциальные коды при представлении каждого из пяти битов в виде электрических или оптических импульсов.

Существование запрещенных комбинаций символов позволяет отбраковывать ошибочные символы, что повышает устойчивость работы сетей 100Base-FX/TX. Так, в Fast Ethernet признаком того, что среда свободна, стала повторяющаяся передача одного из запрещенных для кодирования пользовательских данных символа, а именно символа простоя источника *Idle* (11111). Такой способ позволяет приемнику всегда находиться в синхронизме с передатчиком.

Для отделения кадра Ethernet от символов простоя источника используется комбинация символов начального ограничителя кадра — пара символов *J* (11000) и *K* (10001) кода 4B/5B, а после завершения кадра перед первым символом простоя источника вставляется символ *T* (рис. 13.19).

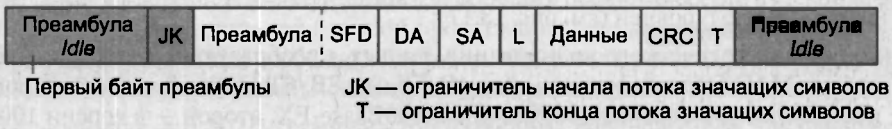


Рис. 13.19. Непрерывный поток данных спецификаций 100Base-FX/TX

После преобразования 4-битных порций кодов MAC в 5-битные порции физического уровня их необходимо представить в виде оптических или электрических сигналов в кабеле, соединяющем узлы сети. В спецификациях 100Base-FX и 100Base-TX для этого используются, соответственно, методы физического кодирования NRZI и MLT-3.

В спецификации 100Base-TX в качестве среды передачи данных используется витая пара UTP категории 5 или STP типа 1. Основным отличием от спецификации 100Base-FX (наряду с методом кодирования MLT-3) является наличие схемы автопереговоров для выбора режима работы порта.

Схема автопереговоров позволяет двум физически соединенным устройствам, которые поддерживают несколько стандартов физического уровня, отличающихся битовой скоростью и количеством витых пар, согласовать наиболее выгодный режим работы. Обычно

процедура автопереговоров происходит при подсоединении сетевого адаптера, который может работать на скоростях 10 и 100 Мбит/с, к концентратору или коммутатору.

Всего в настоящее время определено 5 различных режимов работы, которые могут поддерживать устройства 100Base-TX/T4 на витых парах:

- 10Base-T;
- дуплексный режим 10Base-T;
- 100Base-TX;
- 100Base-T4;
- дуплексный режим 100Base-TX.

Режим 10Base-T имеет самый низкий приоритет в переговорном процессе, а дуплексный режим 100Base-TX — самый высокий.

Переговорный процесс происходит при включении питания устройства, а также может быть инициирован в любой момент модулем управления устройства. Устройство, начавшее процесс автопереговоров, посылает своему партнеру пачку специальных импульсов **FLP** (Fast Link Pulse), в которой содержится 8-битное слово, кодирующее предлагаемый режим взаимодействия, начиная с самого приоритетного, поддерживаемого данным узлом. Импульсы FLP имеют длительность 100 нс, как и импульсы LIT, используемые для тестирования целостности физического соединения в стандарте 10Base-T, однако вместо передачи одного импульса LIT через каждые 16 мс, здесь через тот же интервал передается пачка импульсов FLP.

Если узел-партнер имеет функцию автопереговоров и также способен поддерживать предложенный режим, он отвечает пачкой импульсов FLP, в которой подтверждает этот режим, и на этом переговоры заканчиваются. Если же узел-партнер не может поддерживать запрошенный режим, то он указывает в своем ответе имеющийся в его распоряжении следующий по степени приоритетности режим, и этот режим выбирается в качестве рабочего.

Характеристики производительности Fast Ethernet определяются аналогично характеристикам версии со скоростью Ethernet 10 Мбит/с с учетом неизменного формата кадра, умножения на 10 битовой скорости (в 10 раз больше) и межкадрового интервала (в 10 раз меньше). В результате получаем:

- максимальная скорость протокола в кадрах в секунду (для кадров минимальной длины с полем данных 46 байт) составляет 148 800;
- полезная пропускная способность для кадров минимальной длины равна 54,8 Мбит/с;
- полезная пропускная способность для кадров максимальной длины (поле данных 1500 байт) равна 97,6 Мбит/с.

Gigabit Ethernet

История создания

Достаточно быстро после появления на рынке продуктов Fast Ethernet сетевые интеграторы и администраторы при построении корпоративных сетей почувствовали определенные ограничения. Во многих случаях серверы, подключенные по 100-мегабитному каналу, перегружали магистрали сетей, также работающие на скорости 100 Мбит/с — магистрали FDDI и Fast Ethernet. Ощущалась потребность в следующем уровне иерархии скоростей.

В 1995 году более высокие скорости могли предоставить только коммутаторы ATM, которые из-за высокой стоимости, а также значительных отличий от классических технологий применялись в локальных сетях достаточно редко.

Поэтому логичным выглядел следующий шаг, сделанный IEEE. Летом 1996 года было объявлено о создании группы 802.3z для разработки протокола, в максимальной степени подобного Ethernet, но с битовой скоростью 1000 Мбит/с. Как и в случае Fast Ethernet, сообщение было воспринято сторонниками Ethernet с большим энтузиазмом.

Основной причиной энтузиазма была перспектива плавного перевода сетевых магистралей на Gigabit Ethernet, подобно тому, как были переведены на Fast Ethernet перегруженные сегменты Ethernet, расположенные на нижних уровнях иерархии сети. К тому же опыт передачи данных на гигабитных скоростях уже имелся. В территориальных сетях такую скорость обеспечивала технология SDH, а в локальных — технология Fibre Channel. Последняя используется в основном для подключения высокоскоростной периферии к крупным компьютерам и передает данные по волоконно-оптическому кабелю со скоростью, близкой к гигабитной. (Именно метод кодирования 8B/10B, применяемый в технологии Fiber Channel, был принят в качестве первого варианта физического уровня Gigabit Ethernet.)

Стандарт 802.3z был окончательно принят в 1998 году. Работы по реализации Gigabit Ethernet на витой паре категории 5 были переданы проблемной группе 802.3ab ввиду сложности обеспечения гигабитной скорости на этом типе кабеля, рассчитанного на поддержку скорости 100 Мбит/с. Проблемная группа 802.3ab успешно справилась со своей задачей, и версия Gigabit Ethernet для витой пары категории 5 была принята.

Проблемы совместимости

Основная идея разработчиков стандарта Gigabit Ethernet состояла в максимальном сохранении идей классической технологии Ethernet при достижении битовой скорости в 1000 Мбит/с.

В результате дебатов были приняты следующие решения:

- сохраняются все форматы кадров Ethernet;
- по-прежнему существует полудуплексная версия протокола, поддерживающая метод доступа CSMA/CD;
- поддерживаются все основные виды кабелей, используемых в Ethernet и Fast Ethernet, в том числе волоконно-оптический кабель, витая пара категории 5, экранированная витая пара.

Несмотря на то что в Gigabit Ethernet не стали встраиваться новые функции, поддержание даже достаточно простых функций классического стандарта Ethernet на скорости 1 Гбит/с потребовало решения нескольких сложных задач.

- *Обеспечение приемлемого диаметра сети для работы на разделяемой среде.* В связи с ограничениями, накладываемыми методом CSMA/CD на длину кабеля, версия Gigabit Ethernet для разделяемой среды допускала бы длину сегмента всего в 25 м при сохранении размера кадров и всех параметров метода CSMA/CD неизменными. Так как существует большое количество применений, требующих диаметра сети хотя бы 200 м,

необходимо было каким-то образом решить эту задачу за счет минимальных изменений в технологии Fast Ethernet.

- *Достижение битовой скорости 1000 Мбит/с на оптическом кабеле.* Технология Fibre Channel, физический уровень которой был взят за основу оптоволоконной версии Gigabit Ethernet, обеспечивает скорость передачи данных всего в 800 Мбит/с.
- *Использование в качестве кабеля витой пары.* Такая задача на первый взгляд кажется неразрешимой — ведь даже для 100-мегабитных протоколов требуются достаточно сложные методы кодирования, чтобы уложить спектр сигнала в полосу пропускания кабеля.

Для решения этих задач разработчикам технологии Gigabit Ethernet пришлось внести изменения не только в физический уровень, как это было в случае Fast Ethernet, но и в уровень MAC.

Средства обеспечения диаметра сети в 200 м на разделяемой среде

Для расширения максимального диаметра сети Gigabit Ethernet до 200 м в полудуплексном режиме разработчики технологии предприняли достаточно естественные меры, в основе которых лежало известное соотношение времени передачи кадра минимальной длины и времени оборота (PDV).

Минимальный размер кадра был увеличен (без учета преамбулы) с 64 до 512 байт, или до 4096 бит. Соответственно, время оборота также можно было увеличить до 4095 битовых интервалов, что при использовании одного повторителя сделало допустимым диаметр сети около 200 м.

Для увеличения длины кадра до величины, требуемой в новой технологии, сетевой адаптер должен дополнить поле данных до длины 448 байт так называемым **расширением**, представляющим собой поле, заполненное нулями. Формально минимальный размер кадра не изменился, он по-прежнему равняется 64 байт, или 512 бит, но это объясняется тем, что поле расширения помещается после поля контрольной суммы кадра (FCS). Соответственно, значение этого поля не включается в контрольную сумму и не учитывается при указании длины поля данных в поле длины. Поле расширения является просто расширением сигнала несущей частоты, необходимым для корректного обнаружения коллизий.

Для сокращения накладных расходов в случае использования слишком длинных кадров при передаче коротких квитанций разработчики стандарта разрешили конечным узлам *передавать несколько кадров подряд без передачи среды* другим станциям. Такой режим получил название **режима пульсаций**. Станция может передать подряд несколько кадров с общей длиной не более 65 536 бит, или 8192 байт. При передаче нескольких небольших кадров станции можно не дополнять первый кадр до размера в 512 байт за счет поля расширения, а передавать несколько кадров подряд до исчерпания предела в 8192 байт (в этот предел входят все байты кадра, в том числе преамбула, заголовок, данные и контрольная сумма). Предел 8192 байт называется **длиной пульсации**. Если предел длины пульсации достигается в середине кадра, то кадр разрешается передать до конца. Увеличение «совмещенного» кадра до 8192 байт несколько задерживает доступ к разделяемой среде других станций, но при скорости 1000 Мбит/с эта задержка не столь существенна.

Спецификации физической среды стандарта Gigabit Ethernet

В стандарте 802.3z определены следующие типы физической среды:

- одномодовый волоконно-оптический кабель;
- многомодовый волоконно-оптический кабель 62,5/125;
- многомодовый волоконно-оптический кабель 50/125;
- экранированный сбалансированный медный кабель.

Для передачи данных по традиционному для компьютерных сетей многомодовому волоконно-оптическому кабелю стандарт предписывает применение излучателей, работающих на двух длинах волн: 1300 и 850 нм. Применение светодиодов с длиной волны 850 нм объясняется тем, что они намного дешевле, чем светодиоды, работающие на волне 1300 нм, хотя при этом максимальная длина кабеля уменьшается, так как затухание многомодового оптоволокна на волне 850 м более чем в два раза выше, чем на волне 1300 нм. Тем не менее возможность удешевления чрезвычайно важна для такой в целом дорогой технологии, как Gigabit Ethernet.

Для многомодового оптоволокна стандарт Gigabit Ethernet определяет спецификации 1000Base-SX и 1000Base-LX. В первом случае используется длина волны 850 нм (S означает Short Wavelength), а во втором — 1300 нм (L — Long Wavelength). Спецификация 1000Base-SX разрешает использовать только многомодовый кабель, при этом его максимальная длина составляет около 500 м.

Для спецификации 1000Base-LX в качестве источника излучения всегда применяется полупроводниковый лазер диод с длиной волны 1300 нм. Спецификация 1000Base-LX позволяет работать как с многомодовым (максимальное расстояние до 500 м), так и с одномодовым кабелем (максимальное расстояние зависит от мощности передатчика и качества кабеля и может доходить до нескольких десятков километров).

В качестве среды передачи данных в спецификации 1000-CX определен экранированный сбалансированный медный кабель с волновым сопротивлением 150 Ом. Максимальная длина сегмента составляет всего 25 м, поэтому это решение подходит только для соединения оборудования, расположенного в одной комнате.

Gigabit Ethernet на витой паре категории 5

Как известно, каждая пара кабеля категории 5 имеет гарантированную полосу пропускания до 100 МГц. Для передачи по такому кабелю данных со скоростью 1000 Мбит/с было решено организовать параллельную передачу одновременно по всем четырем парам кабеля.

Это сразу снизило скорость передачи данных по каждой паре до 250 Мбит/с. Однако и для такой скорости необходимо было придумать метод кодирования со спектром, не превышающим 100 МГц. Например, код 4B/5B не позволяет решить поставленную задачу, так как основной вклад в спектр сигнала на такой скорости у него вносит частота 155 МГц. Кроме того, не нужно забывать, что каждая новая версия должна поддерживать не только классический полудуплексный режим, но и дуплексный режим. На первый взгляд кажется, что одновременное использование четырех пар лишает сеть возможности работы в дуплексном режиме, так как не остается свободных пар для одновременной передачи данных в двух направлениях — от узла и к узлу.

Тем не менее проблемная группа 802.3ab нашла решения обеих проблем.

Для кодирования данных был применен код PAM5 с пятью уровнями потенциала: –2, –1, 0, +1, +2. В этом случае за один такт по одной паре передается 2,322 бит информации (log25). Следовательно, для достижения скорости 250 Мбит/с тактовую частоту 250 МГц можно уменьшить в 2,322 раза. Разработчики стандарта решили использовать несколько более высокую частоту, а именно 125 МГц. При этой тактовой частоте код PAM5 имеет спектр уже, чем 100 МГц, то есть он может быть передан без искажений по кабелю категории 5.

В каждом такте передается не 2,322 × 4 = 9,288 бит информации, а 8. Это и дает искомую суммарную скорость 1000 Мбит/с. Передача ровно восьми битов в каждом такте достигается за счет того, что при кодировании информации используются не все 625 (54 = 625) комбинаций кода PAM5, а только 256 (28 = 256). Оставшиеся комбинации приемник задействует для контроля принимаемой информации и выделения правильных комбинаций на фоне шума.

Для организации дуплексного режима разработчики спецификации 802.3ab применили технику выделения принимаемого сигнала из суммарного. Два передатчика работают навстречу друг другу по каждой из четырех пар в одном и том же диапазоне частот (рис. 13.20). H-образная схема гибридной развязки позволяет приемнику и передатчику одного и того же узла использовать одновременно витую пару и для приема, и для передачи (так же, как и в трансиверах Ethernet на коаксиале).

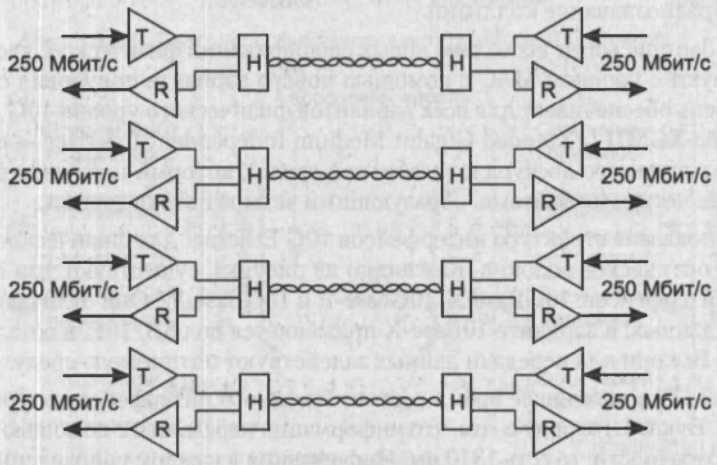


Рис. 13.20. Двунаправленная передача по четырем парам UTP категории 5

Для отделения принимаемого сигнала от собственного приемник вычитает из результирующего сигнала известный ему свой сигнал. Естественно, что это не простая операция и для ее выполнения используются специальные процессоры цифровой обработки сигнала (Digital Signal Processor, DSP).

Вариант технологии Gigabit Ethernet на витой паре расширил *процедуру автопереговоров*, введенную стандартом 100Base-T, за счет включения туда дуплексного и полудуплексного режимов работы на скорости 1000 Мбит/с. Поэтому порты многих коммутаторов Ethernet на витой паре являются универсальными в том смысле, что могут работать на любой из трех скоростей (10, 100 или 1000 Мбит/с).

Характеристики производительности Gigabit Ethernet зависят от того, использует ли коммутатор режим передачи кадров с расширением или же передает их в режиме пульсаций.

В режиме пульсаций на периоде пульсации мы получаем характеристики, в 10 раз отличающиеся от характеристик Fast Ethernet:

- максимальная скорость протокола в кадрах в секунду (для кадров минимальной длины с полем данных 46 байт) составляет 1 488 000;
- полезная пропускная способность для кадров минимальной длины равна 548 Мбит/с;
- полезная пропускная способность для кадров максимальной длины (поле данных 1500 байт) равна 976 Мбит/с.

10G Ethernet

Стандарт **10G Ethernet** определяет только дуплексный режим работы, поэтому он используется исключительно в *коммутируемых* локальных сетях.

Формально этот стандарт имеет обозначение **IEEE 802.3ae** и является поправкой к основному тексту стандарта 802.3. Формат кадра остался неизменным, при этом расширение кадра, введенное в стандарте Gigabit Ethernet, не используется, так как нет необходимости обеспечивать распознавание коллизий.

Стандарт 802.3ae описывает несколько новых спецификаций физического уровня, которые взаимодействуют с уровнем MAC с помощью нового варианта подуровня согласования. Этот подуровень обеспечивает для всех вариантов физического уровня 10G Ethernet единый интерфейс **XGMII** (eXtended Gigabit Medium Independent Interface — расширенный интерфейс независимого доступа к гигабитной среде), который предусматривает параллельный обмен четырьмя байтами, образующими четыре потока данных.

На рис. 13.21 показана структура интерфейсов 10G Ethernet для физического уровня, использующего оптическое волокно. Как видно из рисунка, существуют три группы таких физических интерфейсов: 10GBase-X, 10Gbase-R и 10GBase-W. Они отличаются способом кодирования данных: в варианте 10Base-X применяется код 8B/10B, в остальных двух — код 64B/66B. Все они для передачи данных задействуют оптическую среду.

Группа 10GBase-X в настоящее время состоит из одного интерфейса подуровня PMD — 10GBase-LX4. Буква L говорит о том, что информация передается с помощью волн второго диапазона прозрачности, то есть 1310 нм. Информация в каждом направлении передается одновременно с помощью четырех волн (что отражает цифра 4 в названии интерфейса), которые мультиплексируются на основе техники WDM (рис. 13.22). Каждый из четырех потоков интерфейса XGMII передается в оптическом волокне со скоростью 2,5 Гбит/с.

Максимальное расстояние между передатчиком и приемником стандарта 10GBase-LX4 на многомодовом волокне равно 200–300 м (в зависимости от полосы пропускания волокна), на одномодовом — 10 км.

В каждой из групп 10GBase-W и 10GBase-R может быть три варианта подуровня PMD: S, L и E в зависимости от используемого для передачи информации диапазона волн — 850, 1310 или 1550 нм соответственно. Таким образом, существуют интерфейсы 10GBase-WS, 10GBase-WL, 10GBase-WE и 10GBase-RS, 10GBase-RL и 10GBase-RE. Каждый из них передает информацию с помощью одной волны соответствующего диапазона.

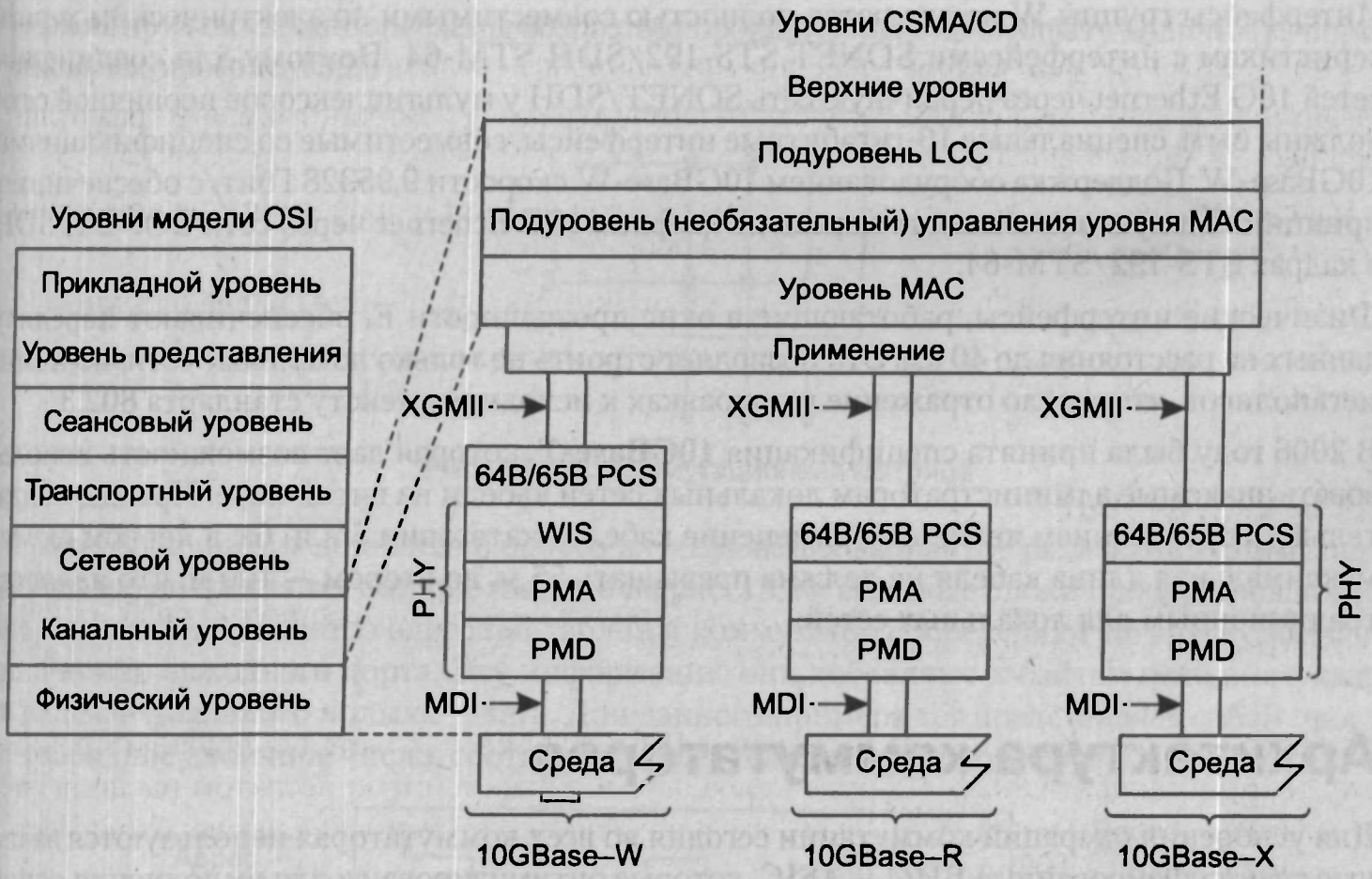


Рис. 13.21. Три группы физических интерфейсов 10G Ethernet

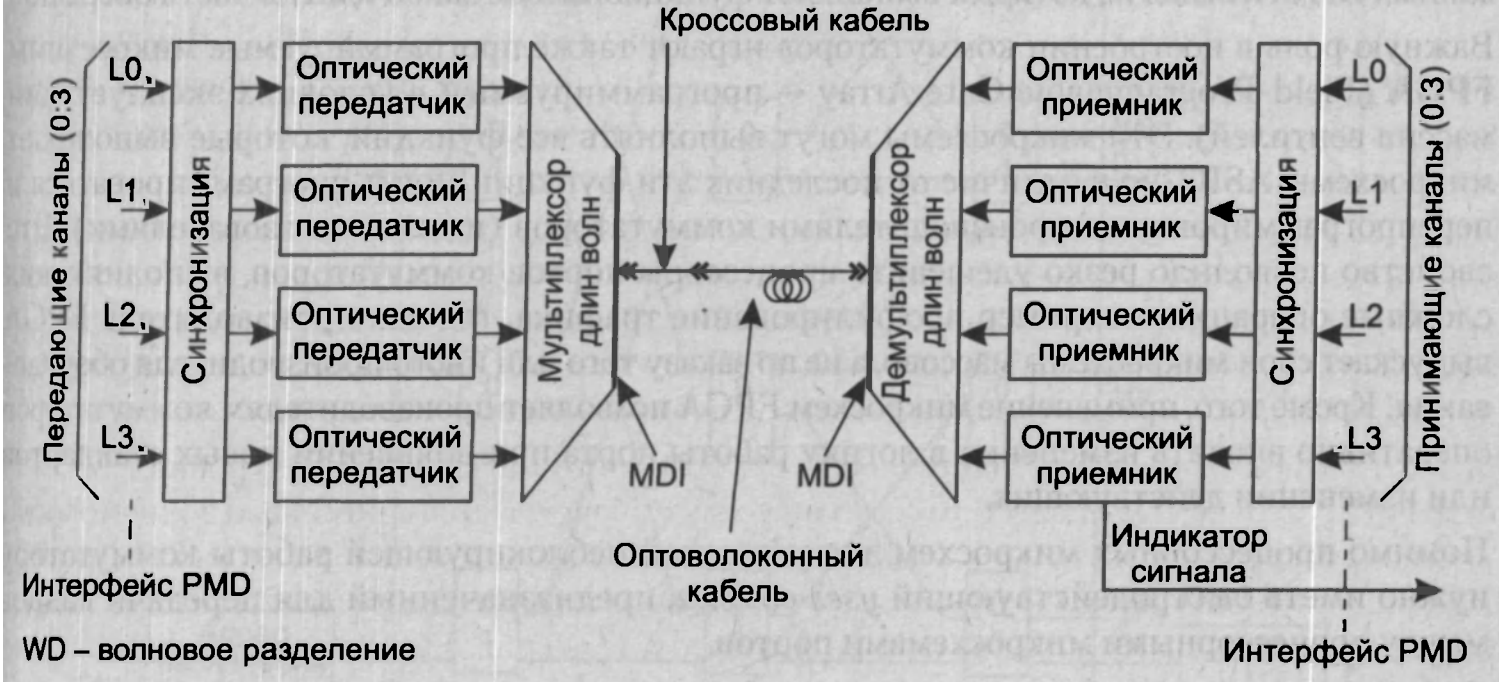


Рис. 13.22. В интерфейсе 10GBase-LX4 используется техника WDM

В отличие от 10GBase-R физические интерфейсы группы 10GBase-W обеспечивают скорость передачи и формат данных, совместимые с интерфейсом SONET STS-192/SDH STM-64. Пропускная способность интерфейсов группы W равна 9,95328 Гбит/с, а эффективная скорость передачи данных — 9,58464 Гбит/с (часть пропускной способности тратится на заголовки кадров STS/STM). Из-за того что скорость передачи информации у этой группы интерфейсов ниже, чем 10 Гбит/с, они могут взаимодействовать только между собой, то есть сое инение, например, интер ейсов 10GBase-RL и 10Base-WL невозможно.

Интерфейсы группы W не являются полностью совместимыми по электрическим характеристикам с интерфейсами SONET STS-192/SDH STM-64. Поэтому для соединения сетей 10G Ethernet через первичную сеть SONET/SDH у мультиплексоров первичной сети должны быть специальные 10-гигабитные интерфейсы, совместимые со спецификациями 10GBase-W. Поддержка оборудованием 10GBase-W скорости 9,95328 Гбит/с обеспечивает принципиальную возможность передачи трафика 10G Ethernet через сети SONET/SDH в кадрах STS-192/STM-64.

Физические интерфейсы, работающие в окне прозрачности E, обеспечивают передачу данных на расстояния до 40 км. Это позволяет строить не только локальные сети, но и сети мегаполисов, что нашло отражение в поправках к исходному тексту стандарта 802.3.

В 2006 году была принята спецификация **10GBase-T**, которая дает возможность использовать знакомые администраторам локальных сетей кабели на витой паре. Правда, обязательным требованием является применение кабелей категории 6 или 6a: в первом случае максимальная длина кабеля не должна превышать 55 м, во втором — 100 м, что является традиционным для локальных сетей.

Архитектура коммутаторов

Для ускорения операций коммутации сегодня во всех коммутаторах используются заказные специализированные БИС — ASIC, которые оптимизированы для выполнения основных операций коммутации. Часто в одном коммутаторе имеется несколько специализированных БИС, каждая из которых выполняет функционально законченную часть операций.

Важную роль в построении коммутаторов играют также программируемые микросхемы **FPGA** (Field-Programmable Gate Array — программируемый в условиях эксплуатации массив вентилей). Эти микросхемы могут выполнять все функции, которые выполняют микросхемы ASIC, но в отличие от последних эти функции могут программироваться и перепрограммироваться производителями коммутаторов (и даже пользователями). Это свойство позволило резко удешевить процессоры портов коммутаторов, выполняющих сложные операции, например профилирование трафика, так как производитель FPGA выпускает свои микросхемы массово, а не по заказу того или иного производителя оборудования. Кроме того, применение микросхем FPGA позволяет производителям коммутаторов оперативно вносить изменения в логику работы порта при появлении новых стандартов или изменении действующих.

Помимо процессорных микросхем для успешной неблокирующей работы коммутатору нужно иметь быстродействующий *узел обмена*, предназначенный для передачи кадров между процессорными микросхемами портов.

В настоящее время в коммутаторах узел обмена строится на основе одной из трех схем:

- коммутационная матрица;
- общая шина;
- разделяемая многовходовая память.

Часто эти три схемы комбинируются в одном коммутаторе.

Коммутационная матрица обеспечивает наиболее простой способ взаимодействия процессоров портов, и именно этот способ был реализован в первом промышленном коммутаторе локальных сетей. Однако реализация матрицы возможна только для определенного числа

портов, причем сложность схемы возрастает пропорционально квадрату количества портов коммутатора (рис. 13.23).

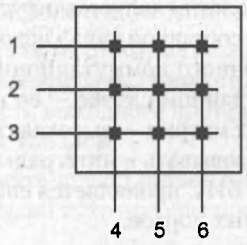


Рис. 13.23. Коммутационная матрица

Более детальное представление одного из возможных вариантов реализации коммутационной матрицы для восьми портов дано на рис. 13.24. Входные блоки процессоров портов на основании просмотра адресной таблицы коммутатора определяют по адресу назначения номер выходного порта. Эту информацию они добавляют к байтам исходного кадра в виде специального ярлыка — тега. Для данного примера тег представляет собой просто 3-разрядное двоичное число, соответствующее номеру выходного порта.

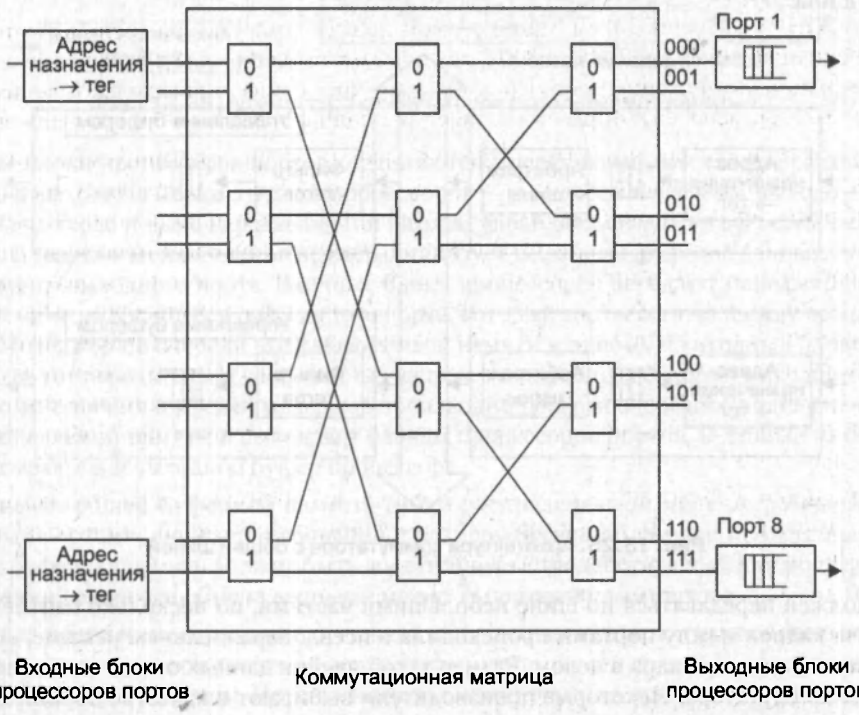


Рис. 13.24. Реализация коммутационной матрицы 8 × 8 с помощью двоичных переключателей

Матрица состоит из трех уровней двоичных переключателей, которые соединяют свой вход с одним из двух выходов в зависимости от значения бита тега. Переключатели первого уровня управляются первым битом тега, второго — вторым, а третьего — третьим.

Матрица может быть реализована и иначе, на основании комбинационных схем другого типа, но ее особенностью все равно остается технология коммутации физических каналов. Известным недостатком этой технологии является отсутствие буферизации данных внутри коммутационной матрицы — если составной канал невозможно построить из-за занятости выходного порта или промежуточного коммутационного элемента, то данные должны накапливаться в их источнике, в данном случае — во входном блоке порта, принявшего кадр. Основные достоинства таких матриц — высокая скорость коммутации и регулярная структура, которую удобно реализовывать в интегральных микросхемах. Зато после реализации матрицы $N \times N$ в составе БИС проявляется еще один ее недостаток — сложность наращивания числа коммутируемых портов.

В **коммутаторах с общей шиной** процессоры портов связывают высокоскоростной шиной, используемой в режиме разделения времени.

Пример такой архитектуры приведен на рис. 13.25. Чтобы шина не блокировала работу коммутатора, ее производительность должна равняться, по крайней мере, сумме производительностей всех портов коммутатора. Для модульных коммутаторов характерно то, что путем удачного подбора модулей с низкоскоростными портами можно обеспечить неблокирующий режим работы, но в то же время некоторые сочетания модулей с высокоскоростными портами могут приводить к структурам, у которых узким местом является общая шина.

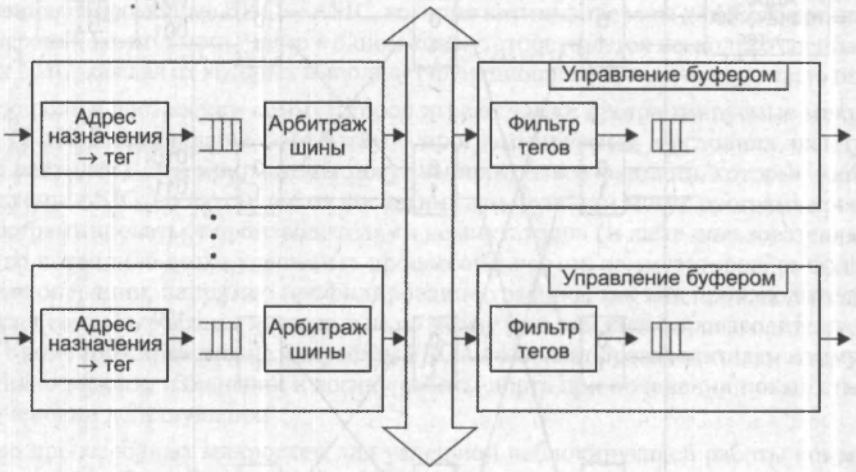


Рис. 13.25. Архитектура коммутатора с общей шиной

Кадр должен передаваться по шине небольшими частями, по несколько байтов, чтобы передача кадров между портами происходила в псевдопараллельном режиме, не внося задержек в передачу кадра в целом. Размер такой ячейки данных определяется производителем коммутатора. Некоторые производители выбирают в качестве порции данных, переносимых по шине за одну операцию, ячейку ATM с ее полем данных в 48 байт. Такой подход облегчает трансляцию протоколов локальных сетей в протокол ATM, если коммутатор поддерживает эти технологии. Кроме того, небольшой размер ячейки (ее формат может быть и фирменным, так как перенос данных между портами является сугубо внутренней операцией) уменьшает задержки доступа порта к общей шине.

Входной блок процессора помещает в ячейку, переносимую по шине, тег, в котором указывает номер порта назначения. Каждый выходной блок процессора порта содержит фильтр тегов, который выбирает теги, предназначенные данному порту.

Шина, так же как и коммутационная матрица, не может осуществлять промежуточную буферизацию, но поскольку данные кадра разбиваются на небольшие ячейки, задержек с начальным ожиданием доступности выходного порта в такой схеме нет — здесь работает принцип коммутации пакетов, а не каналов.

Разделяемая многовходовая память представляет собой третью базовую архитектуру взаимодействия портов. Пример такой архитектуры приведен на рис. 13.26.

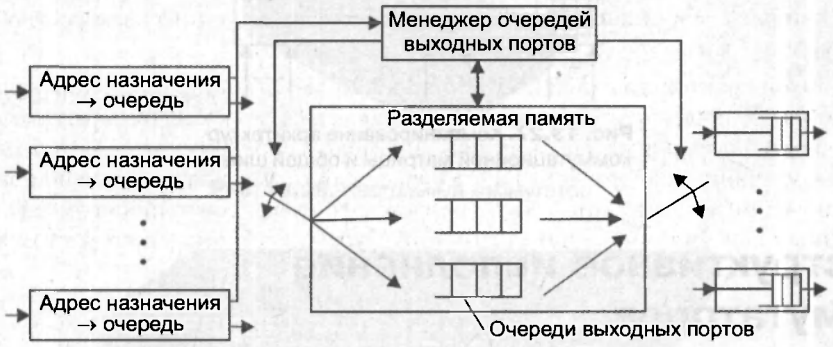


Рис. 13.26. Архитектура коммутаторов с разделяемой памятью

Входные блоки процессоров портов соединяются с переключаемым входом разделяемой памяти, а выходные блоки этих же процессоров — с ее переключаемым выходом. Переключением входа и выхода разделяемой памяти управляет *менеджер очередей выходных портов*. В разделяемой памяти менеджер организует несколько очередей данных, по одной для каждого выходного порта. Входные блоки процессоров передают менеджеру портов запросы на запись данных в очередь того порта, который соответствует адресу назначения кадра. Менеджер по очереди подключает вход памяти к одному из входных блоков процессоров и тот переписывает часть данных кадра в очередь определенного выходного порта. По мере заполнения очередей менеджер производит также поочередное подключение выхода разделяемой памяти к выходным блокам процессоров портов, и данные из очереди переписываются в выходной буфер процессора.

Применение общей буферной памяти, гибко распределяемой менеджером между отдельными портами, снижает требования к размеру буферной памяти процессора порта. Однако буферная память должна быть достаточно быстродействующей для поддержания необходимой скорости обмена данными между N портами коммутатора.

Комбинированные коммутаторы. У каждой из описанных архитектур есть свои достоинства и недостатки, поэтому часто в сложных коммутаторах эти архитектуры применяются в комбинации друг с другом. Пример такого комбинирования приведен на рис. 13.27.

Коммутатор состоит из модулей с фиксированным количеством портов (2–12), выполненных на основе специализированной БИС, реализующей архитектуру коммутационной матрицы. Если порты, между которыми нужно передать кадр данных, принадлежат одному модулю, то передача кадра осуществляется процессорами модуля на основе имеющейся в модуле коммутационной матрицы. Если же порты принадлежат разным модулям, то про-

цессоры общаются по общей шине. В такой архитектуре передача кадров внутри модуля будет происходить быстрее, чем при межмодульной передаче, так как коммутационная матрица — это наиболее быстрое, хотя и наименее масштабируемое средство взаимодействия портов. Скорость внутренней шины коммутаторов может достигать нескольких гигабит в секунду, а у наиболее мощных моделей — до нескольких десятков гигабит в секунду.

Рис. 13.27. Комбинирование архитектур коммутационной матрицы и общей шины

Конструктивное исполнение коммутаторов

На конструктивное исполнение коммутаторов большое влияние оказывает их область применения. Настольные коммутаторы и коммутаторы рабочих групп чаще всего выпускаются как устройства с фиксированным количеством портов, корпоративные коммутаторы — как модульные устройства на основе шасси, а коммутаторы отделов могут иметь стековую конструкцию. Такое деление не является жестким, и в качестве корпоративного коммутатора может использоваться, например, стековый коммутатор.

Коммутатор с фиксированным количеством портов — это наиболее простое конструктивное исполнение, когда устройство представляет собой отдельный корпус со всеми необходимыми элементами (портами, органами индикации и управления, блоком питания), и эти элементы заменять нельзя.

Настольные коммутаторы представляют собой наиболее простой тип устройств с фиксированным количеством портов (рис. 13.28). Обычно все порты такого коммутатора поддерживают одну среду передачи, общее количество портов изменяется от 4 до 48. Порты такого коммутатора являются чаще всего интерфейсами 10/100 или 10/100/1000 Мбит/с на витой паре, поддерживающими автопереговоры. Как правило, такой коммутатор не поддерживает удаленное управление по протоколу SNMP.

Коммутатор рабочей группы с фиксированным количеством портов (рис. 13.29) имеет, как правило, множество портов для подключения пользовательских компьютеров — как и у настольного коммутатора, эти порты обычно являются интерфейсами 10/100 или 10/100/1000 Мбит/с на витой паре, поддерживающими автопереговоры. В нашем примере коммутатор оснащен 24 портами 10/100 Мбит/с Кроме того, такой коммутатор имеет несколько магистральных портов для соединения с коммутаторами верхних уровней.

В нашем примере коммутатор имеет 4 магистральных порта, но они выполнены в особом конструктивном исполнении как слоты для установки модулей портов стандарта SFP.

Дело в том, что начиная со стандарта Gigabit Ethernet, порты для работы на оптическом волокне начали выпускаться в виде отдельных модулей, устанавливаемых в специальные слоты коммуникационных устройств. Такая конструкция позволяет легко переходить от одного типа оптического волокна к другому, например от многомодового к одномодовому, путем замены модуля порта. Существует два популярных стандарта на конструктивное исполнение модулей портов Gigabit Ethernet и их интерфейс с самим устройством: GBIC и SFP (рис. 13.30).

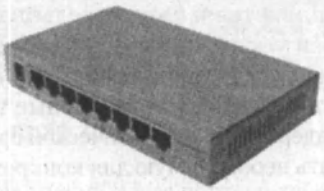


Рис. 13.28. Настольный коммутатор

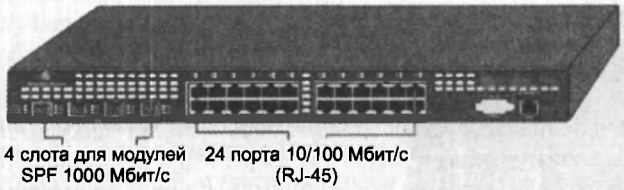


Рис. 13.29. Коммутатор рабочей группы с магистральными портами

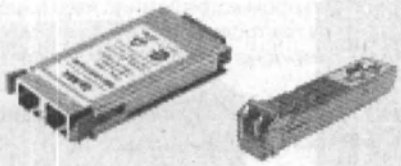


Рис. 13.30. Модули GBIC (*слева*) и SFP (*справа*)

Оба эти стандарта приняты комитетом SFF (Small Form Factor committee — Комитет производителей компактного оборудования), который был образован в 1990 году как консорциум производителей периферийного оборудования для компьютеров, а затем расширил свои функции. Стандарты SFF являются результатом взаимной договоренности между производителями оборудования. Модули GBIC (Gigabit Ethernet Interface Converter — конвертор интерфейса Gigabit Ethernet) появились раньше, они обладают большими размерами, чем модули SFP (Small Factor Pluggable module — устанавливаемый модуль небольшого размера), которые были стандартизованы позднее. Модули SFP называют также моделями мини-GBIC. Несмотря на то что изначально и модули GBIC, и модули SFP были задуманы

как сменная часть портов Gigabit Ethernet для оптического волокна, выпускаются модули SFP и для витой пары, так как это делает слоты SFP коммутаторов (и маршрутизаторов) универсальными.

В том случае, если коммутатор рабочей группы поддерживает интерфейсы 10G Ethertnet (их нет у коммутатора на рис. 13.29), они также выполняются как слоты с устанавливаемыми модулями. Существует несколько стандартов таких модулей: XENPAK, XSP и SFP+ (последний вариант самый компактный). Все эти стандарты представляют собой результат взаимной договоренности между производителями оборудования.

Модульный коммутатор выполняется в виде отдельных модулей с фиксированным количеством портов, эти модули устанавливаются на общее шасси (рис. 13.31). Шасси имеет внутреннюю шину для объединения отдельных модулей в единое устройство. Для модульного коммутатора могут существовать различные типы модулей, отличающиеся количеством портов и типом поддерживаемой физической среды. Модульные коммутаторы позволяют более точно подобрать необходимую для конкретного применения конфигурацию коммутатора, а также гибко и с минимальными затратами реагировать на изменения конфигурации сети.

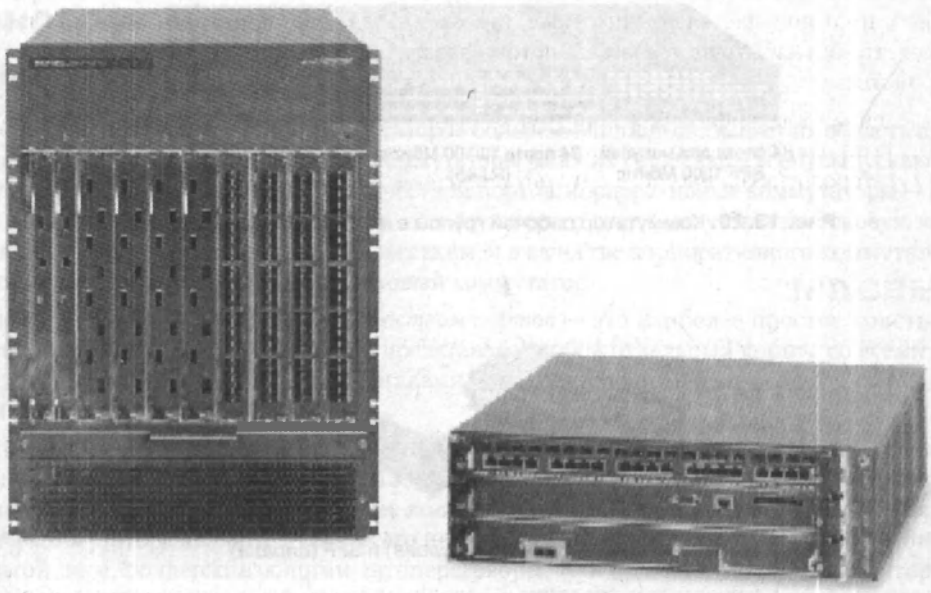


Рис. 13.31. Модульные коммутаторы на основе шасси

Ввиду ответственной работы, которую выполняют модульные коммутаторы, они снабжаются модулем управления, системой терморегулирования, избыточными источниками питания и возможностью замены модулей «на лету».

Недостатком коммутатора на основе шасси является высокая начальная стоимость такого устройства для случая, когда предприятию на первом этапе создания сети нужно установить всего 1–2 модуля. Высокая стоимость шасси вызвана тем, что оно поставляется вместе со всеми общими устройствами, такими как избыточные источники питания и т. п.

Стековый коммутатор, как и коммутатор с фиксированным числом портов, выполнен в виде отдельного корпуса без возможности замены отдельных его модулей. Несколько типичных стековых коммутаторов Ethernet показаны на рис. 13.32.

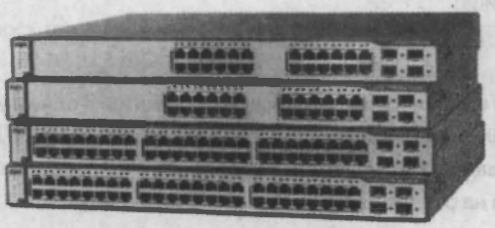


Рис. 13.32. Стековые коммутаторы Ethernet

Стековые коммутаторы имеют специальные порты и кабели для объединения нескольких корпусов в единый коммутатор с общим блоком управления. Стековые коммутаторы могут поддерживать различные физические среды передачи, что делает их почти такими же гибкими, как модульные концентраторы, но при этом стоимость этих устройств в расчете на один порт получается обычно ниже, так как сначала предприятие может купить одно устройство без избыточного шасси, а потом нарастить стек еще несколькими аналогичными устройствами.

Приведенная классификация конструктивного исполнения справедлива не только для коммутаторов, но и для коммуникационных устройств всех типов — маршрутизаторов, коммутаторов глобальных сетей, мультиплексоров SDH/OTN/DWDM.

Выводы

Для логической структуризации сети применяются мосты и их современные преемники — коммутаторы локальных сетей. Устройства обоих типов работают на основе одного и того же стандарта IEEE 802.1D, но коммутаторы обладают гораздо более высоким быстродействием за счет параллельной обработки потоков данных.

Коммутаторы являются самообучающимися устройствами, так как строят таблицы продвижения автоматически на основе слежения за передаваемыми кадрами.

Недостатком коммутаторов является невозможность работы в сетях с петлевидными связями. Другим недостатком сетей, построенных на коммутаторах, является отсутствие защиты от широковещательного шторма.

Применение коммутаторов позволяет сетевым адаптерам использовать дуплексный режим работы. В этом режиме отсутствует этап доступа к разделяемой среде, а общая скорость передачи данных удваивается.

В дуплексном режиме для борьбы с перегрузками коммутаторов используется метод обратной связи, описанный в стандарте 802.3x. Он позволяет приостановить на некоторое время поступление кадров от непосредственных соседей перегруженного коммутатора.

Основными характеристиками производительности коммутатора являются: скорость фильтрации кадров, скорость продвижения кадров, общая пропускная способность по всем портам в мегабитах в секунду, задержка передачи кадра.

Потребности в высокоскоростной и в то же время недорогой технологии для подключения к сети мощных рабочих станций привели к созданию нескольких скоростных версий Ethernet: Fast Ethernet

со скоростью 100 Мбит/с, Gigabit Ethernet со скоростью 1 Гбит/с и 10G Ethernet со скоростью 10 Гбит/с.

Существует несколько основных вариантов внутренней архитектуры коммутатора, в основе которых лежит:

- коммутационная матрица;
- разделяемая память;
- общая шина.

Кроме того, применяется комбинирование основных вариантов в одном устройстве.

По конструктивному исполнению коммутаторы разделяются на:

- устройства с фиксированным количеством портов;
- модульные устройства на основе шасси;
- стековые коммутаторы.

Вопросы и задания

1. Что из перечисленного можно отнести к недостаткам сетей на разделяемой среде:
   а) неопределенная доля пропускной способности, приходящаяся на один узел сети;
   б) сложность подключения нового узла к сети;
   в) плохая масштабируемость;
   г) сложность организации широковещания.
2. Почему мост, работающий в соответствии со стандартом IEEE 802.1D, называют «прозрачным»? Варианты ответов:
   а) потому что он передает кадры Ethernet без изменения;
   б) потому что конечные узлы «не замечают» его присутствия в сети;
   в) потому что мост строит таблицу продвижения автоматически.
3. На основе изучения каких адресов автоматически строится таблица продвижения моста? Варианты ответов:
   а) MAC-адресов назначения; б) MAC-адресов источника.
4. К каким негативным последствиям приводит наличие петель в сети, построенной на коммутаторах, работающих в соответствии с алгоритмом прозрачного моста? Варианты ответов:
   а) кадры могут дублироваться;
   б) кадры могут зацикливаться;
   в) таблица продвижения может постоянно перестраиваться.
5. Для какой цели записи таблицы продвижения имеют ограниченный срок жизни?
6. Может ли скорость продвижения превосходить скорость фильтрации?
7. Чем коммутатор отличается от моста? Варианты ответов:
   а) количеством портов;
   б) способом построения таблицы продвижения;
   в) дополнительными функциями;

г) производительностью.

8. При каком распределении трафика неблокирующий коммутатор с 12-ю портами Fast Ethernet и одним портом Gigabit Ethernet оправдывает свое название? Варианты ответов:
   а) входной трафик всех портов Fast Ethernet, которые работают с близкой к 100 % нагрузкой, направлен в порт Gigabit Ethernet;
   б) входной трафик порта Gigabit Ethernet, который работает с близкой к 100 % нагрузкой, равномерно распределен между 12-ю портами Fast Ethernet;
   в) входной трафик всех портов Fast Ethernet, которые работают с 50-процентной нагрузкой, направлен в порт Gigabit Ethernet.
9. Какие механизмы коммутаторы используют для борьбы с перегрузками в дуплексном режиме работы? Варианты ответов:
   а) обратное давление;
   б) сообщение PAUSE;
   в) динамическое увеличение скорости порта.
10. К каким последствиям может привести недостаточный объем памяти, выделенной под таблицу продвижения коммутатора? Варианты ответов:
   а) постоянная перестройка таблицы продвижения;
   б) затопление сети кадрами с неизученным адресом назначения;
   в) потеря кадров.
11. Совпадают ли форматы кадров 10 Мбит/с Ethernet и Fast Ethernet?
12. Для какой цели в формат кадра Gigabit Ethernet было введено поле расширения? Варианты ответов:
   а) для повышения производительности сети;
   б) для передачи дополнительных адресов назначения;
   в) для увеличения максимального диаметра сегмента разделяемой среды.
13. Может ли в технологии 10G Ethernet использоваться разделяемая среда?
14. Поддерживается ли режим автопереговоров для волоконно-оптических портов?
15. Какой особенности физического интерфейса соответствует цифра 4 в спецификации 10GBase-LX4?
16. Можно ли коммутатор локальной сети с интерфейсом 10GBase-WL непосредственно присоединить к порту STM-64 мультиплексора SDH?

ГЛАВА 14 Интеллектуальные функции коммутаторов

Коммутируемые сети гораздо более производительны и масштабируемы, чем сети на разделяемой среде. Тем не менее локальная сеть, коммутаторы которой поддерживают алгоритм прозрачного моста, по-прежнему обладает рядом принципиальных недостатков. Прежде всего, остается нерешенной проблема надежности сети, так как древовидная топология коммутируемых локальных сетей очень уязвима — отказ любой линии связи или коммутатора приводит к потере связности сети, сеть фактически распадается на два или более сегмента.

Кроме того, такая сеть не имеет барьеров на пути ошибочного трафика, генерируемого любым из ее узлов, так как алгоритм прозрачного моста подразумевает передачу кадров с неизученным или широковещательным адресом всем узлам сети. Примером такой нежелательной ситуации является широковещательный шторм, возникающий из-за неисправности всего одного сетевого адаптера и приводящий к потере работоспособности всей сети. Говорят, что сеть на коммутаторах является «плоской», поскольку такая сеть не имеет барьеров на пути нежелательного трафика.

Ограничения древовидной топологии преодолеваются с помощью интеллектуальных функций коммутаторов, которые наделяют локальные сети дополнительными возможностями. Так, в коммутируемых локальных сетях широко применяется протокол покрывающего дерева (STP), который за счет резервных связей в сети автоматически находит новый вариант древовидной топологии при отказах и тем самым обеспечивает отказоустойчивость сети. Алгоритм покрывающего дерева был разработан одновременно с алгоритмом прозрачного моста (то есть в начале 80-х) и с тех пор успешно применяется в локальных сетях; последняя версия этого алгоритма Rapid STP позволила значительно сократить время перехода сети на резервную топологию.

Техника виртуальных локальных сетей (VLAN) позволяет разбивать коммутируемую локальную сеть на несколько обособленных логических сегментов, предотвращающих распространение нежелательного трафика по всей сети, кроме того, это свойство улучшает управляемость сети. Обособленные сегменты виртуальных локальных сетей затем могут быть соединены в составную сеть уже с помощью маршрутизаторов, при этом благодаря программному делению сети на сегменты очень удобно быстро поменять структуру сети.

Механизм агрегирования линий связи позволяет объединить несколько линий связи (физических каналов) в один логический канал. Это повышает как производительность, так и надежность сети. Агрегирование линий связи полезно в тех случаях, когда 10-кратное повышение скорости какой-нибудь связи за счет перехода на более высокий уровень иерархии протокола Ethernet либо невозможно (например, из-за того, что существующая скорость линии является предельной, каковой на момент написания этой книги была скорость 10 Гбит/с), либо экономически или организационно менее выгодно, чем параллельное использование нескольких имеющихся портов.

Новые развитые возможности коммутаторов локальных сетей обеспечивают поддержку методов QoS для различных типов трафика, включая приоритетные и взвешенные очереди, обратную связь, резервирование ресурсов.

Алгоритм покрывающего дерева

В коммутируемых локальных сетях проблема обеспечения надежности сети имеет свою специфику: базовый протокол прозрачного моста корректно работает только в сети с *древовидной топологией*, в которой между любыми двумя узлами сети существует единственный маршрут. Тем не менее очевидно, что для надежной работы сети необходимо наличие альтернативных маршрутов между узлами, которые можно использовать при отказе основного маршрута. Наиболее простым решением этой проблемы является построение сети с альтернативными маршрутами, ручное нахождение связной древовидной топологии и ручное блокирование (то есть перевод в административное состояние «отключен») всех портов, которые не входят в найденную топологию. В случае отказа сети этот процесс должен повторяться, опять же в ручном режиме. Понятно, что надежность сети в этом случае оказывается не очень высокой, так как время пребывания ее в неработоспособном состоянии будет исчисляться минутами: сначала нужно обнаружить отказ и локализовать его (то есть не только зафиксировать факт, что в сети что-то перестало работать, но и понять, какая именно связь пострадала и требует обхода), затем найти новый работоспособный вариант топологи сети (если он, конечно, существует), а потом его сконфигурировать.

Для автоматического выполнения перечисленных действий, то есть нахождения и конфигурирования активной древовидной топологии, мониторинга состояния ее связей и перехода к новой древовидной топологии при обнаружении отказа связи в коммутируемых локальных сетях используются **алгоритм покрывающего дерева** (Spanning Tree Algorithm, STA) и реализующий его **протокол покрывающего дерева** (Spanning Tree Protocol, STP).

Алгоритм покрывающего дерева, разработанный достаточно давно, в 1983 году, был признан IEEE удачным решением и включен в ту же спецификацию 802.1D, в которой описывается и сам алгоритм прозрачного моста. Сегодня протокол STP широко применяется в наиболее массовых устройствах современных локальных сетей — коммутаторах. Протокол STP обновлялся несколько раз, последняя его редакция описана в документе 802.1D-2004; новая версия протокола получила название RSTP (Rapid STP, то есть быстрый протокол покрывающего дерева), так как предыдущие версии STP недостаточно быстро находили новую древовидную топологию — на это могло уйти до 50 секунд. Новая версия протокола покрывающего дерева — RSTP — работает значительно быстрее, затрачивая на поиск новой топологи несколько секунд.

Мы сначала рассмотрим классическую версию STP, а затем ее быстрый вариант — RSTP.

Классическая версия STP

Протокол STP формализует сеть (рис. 14.1, *а*) в виде графа (рис. 14.1, *б*), вершинами которого являются коммутаторы и сегменты сети.

Сегмент — это связная часть сети, не содержащая коммутаторов (и маршрутизаторов). Сегмент может быть разделяемым (во время создания алгоритма STA это был единственный тип сегмента) и включать устройства физического уровня — повторители/концентраторы, существование которых коммутатор, будучи устройством канального уровня, «не замечает». Сегмент также может представлять собой двухточечный канал, в коммутируемых локальных сетях это единственный тип сегмента.

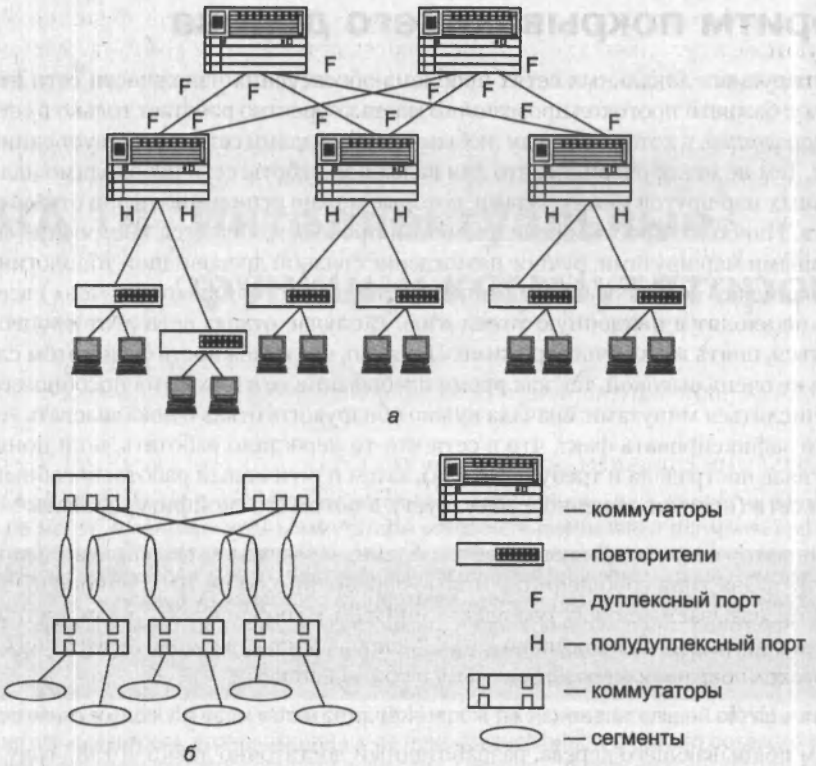


Рис. 14.1. Формализованное представление сети в соответствии с алгоритмом STA

Протокол покрывающего дерева обеспечивает построение древовидной топологии связей с единственным путем минимальной длины от каждого коммутатора и от каждого сегмента до некоторого выделенного **корневого коммутатора** — корня дерева. *Единственность* пути гарантирует отсутствие петель, а *минимальность* расстояния — рациональность маршрутов следования трафика от периферии сети к ее магистрали, роль которой исполняет корневой коммутатор.

В качестве расстояния в STA используется **метрика** — традиционная для протоколов маршрутизации величина, обратно пропорциональная пропускной способности сегмента. В STA метрика определяется также как *условное время передачи бита сегментом*. В версии 802.1D-1998 эта величина является 16-разрядной, а в версии 802.1D-2004 — 32-разрядной.

В версии 1998 года выбраны следующие значения метрики: 10 Мбит/с — 100, 100 Мбит/с — 19, 1 Гбит/с — 4, 10 Гбит/с — 2. В текущей версии 802.1D-2004 используются такие значения метрик, которые расширяют диапазон скоростей сегментов до 10 Тбит/с (то есть с большим запасом относительно сегодняшнего уровня максимальной для Ethernet скорости в 10 Гбит/с), давая такому сегменту значение 2; соответственно сегмент 100 Гбит/с получает значение 200, 10 Гбит/с — 2000, 1 Гбит/с — 20 000, 100 Мбит/с — 200 000, а 10 Мбит/с — 2 000 000.

Идентификатор коммутатора — это 8-байтовое число, шесть младших байтов которого составляют MAC-адрес его блока управления, отрабатывающего алгоритм STA (напомним,

что портам коммутаторов и мостов для выполнения своей основной функции MAC-адреса не требуются), а два старших байта называются приоритетом коммутатора (значение по умолчанию равно 32 768) и конфигурируются вручную, что, как мы увидим далее, позволяет администратору сети влиять на процесс выбора корневого коммутатора.

Корневой порт коммутатора — это порт, который имеет кратчайшее расстояние до корневого коммутатора (точнее, до любого из портов корневого коммутатора).

Идентификатором порта служит 2-байтовое число. Младший байт содержит порядковый номер данного порта в коммутаторе, а значение старшего байта является приоритетом (значение по умолчанию равно 128) и задается администратором.

Назначенным коммутатором сегмента объявляется коммутатор, у которого расстояние до корневого коммутатора является минимальным.

Назначенный порт — это порт назначенного коммутатора сегмента, подключенный к данному сегменту.

Протокольными единицами данных моста (Bridge Protocol Data Unit, BPDU) называются специальные пакеты, которыми периодически обмениваются коммутаторы для автоматического определения конфигурации дерева. Пакеты BPDU переносят данные об идентификаторах коммутаторов и портов, а также о расстоянии до корневого коммутатора. Существует два типа сообщений, которые переносят пакеты BPDU: конфигурационные сообщения, называемые также сообщениями Hello, и сообщения с уведомлениями об изменении конфигурации. Для доставки BPDU используется групповой адрес 01:80:C2:00:00:00, позволяющий организовать эффективный обмен данными.

Интервал Hello — это интервал между генерацией сообщений Hello; он настраивается администратором и обычно составляет от 1 до 4 секунд; по умолчанию — 2 секунды.

Три этапа построения дерева

На рис. 14.2 приведен пример сети из стандарта 802.1D-2004, который иллюстрирует работу протокола STP. Мы также будем использовать этот пример в своем описании.

В этом примере сеть построена на восьми коммутаторах, которые имеют идентификаторы со значениями от 111 до 888 (для удобства записи здесь используются сокращенные до 3-х разрядов значения MAC-адресов коммутаторов). Все коммутаторы соединены друг с другом двухточечными связями, которые образуют сегменты *A*–*N*. Порты 3 и 4 коммутаторов с 555 по 888 соединены с конечными узлами сети, то есть компьютерами (на рисунке не показаны). Все связи в сети — это связи со скоростью 100 Мбит/с (Fast Ethernet).

Алгоритм STA определяет активную конфигурацию сети за три этапа.

Первый этап — определение корневого коммутатора, от которого строится дерево.

В качестве корневого коммутатора выбирается коммутатор с *наименьшим значением идентификатора*. В исходном состоянии каждый коммутатор считает себя корневым, поэтому он генерирует и передает своим соседям сообщения Hello, в которых помещает свой идентификатор в качестве идентификатора корневого коммутатора. Как только коммутатор получает от соседа сообщение Hello, в котором содержится идентификатор корневого коммутатора, меньший его собственного, он перестает считать себя корневым коммутатором и генерировать свои сообщения Hello, но начинает ретранслировать сообщения Hello, получаемые от соседей.

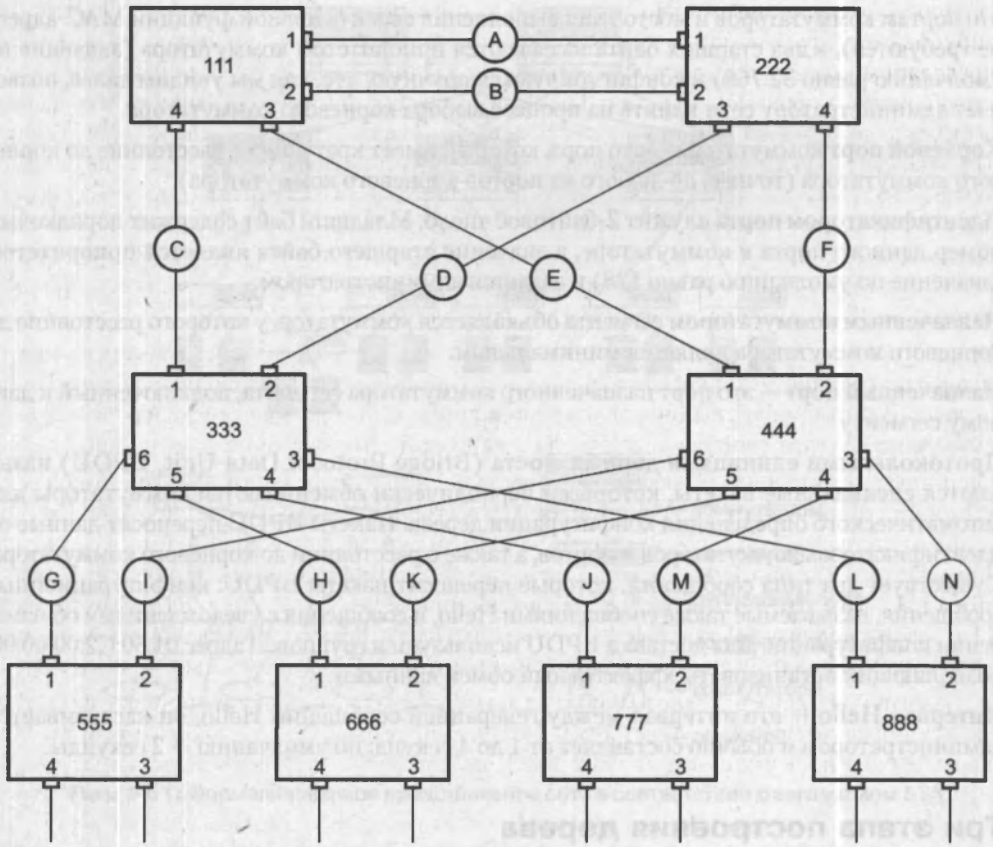


Рис. 14.2. Пример сети, иллюстрирующей работу STP

Если администратор не вмешается в этот процесс, корневой коммутатор выбирается достаточно случайным образом — им станет устройство с минимальным MAC-адресом блока управления. Очевидно, что такой выбор может оказаться далеко не рациональным. Например, при выборе коммутатора 555 в качестве корневого значительная часть трафика проходила бы через большое количество транзитных сегментов и коммутаторов. Поэтому пускать данный процесс «на самотек» администратору не стоит — лучше в него вмешаться и назначить корневой коммутатор осознанно (путем соответствующего конфигурирования старших байтов идентификаторов коммутатора), чтобы выбранный коммутатор действительно занимал центральное место в соединениях сегментов.

В нашем примере мы предполагаем, что администратор не стал менять приоритеты коммутаторов, так что у всех коммутаторов они остались равными значению 32 768 (значение по умолчанию), и корневым коммутатором стал коммутатор с идентификатором 111.

Второй этап — выбор корневого порта для каждого коммутатора.

Корневым портом коммутатора является тот порт, расстояние от которого до корневого коммутатора является минимальным. Сам корневой коммутатор корневых портов не имеет.

Для определения корневого порта каждый коммутатор использует пакеты Hello, ретранслируемые ему другими коммутаторами. На основании этих пакетов каждый коммутатор определяет минимальные расстояния от всех своих портов до корневого коммутатора. При ретрансляции сообщения Hello каждый коммутатор увеличивает указанное в сообщении расстояние до корня на метрику того сегмента, из которого принят данный пакет. Тем самым в пакете Hello по мере прохождения через коммутаторы наращивается поле, показывающее расстояние до корневого коммутатора. Например, если считать, что все сегменты в рассматриваемом примере являются сегментами Ethernet со скоростью 100 Мбит/с, то коммутатор 222, приняв из сегмента *A* пакет Hello со значением расстояния, равным 0, увеличивает его на 200 000 условных единиц (если коммутатор работает с величинами метрики, рекомендованными версией стандарта STP от 2004 года).

Ретранслируя пакеты, каждый коммутатор для каждого своего порта запоминает минимальное расстояние до корня, встретившееся во всех принятых этим портом пакетах Hello. По завершении процедуры определения конфигурации покрывающего дерева каждый коммутатор находит свой корневой порт (с минимальным расстоянием до корня).

При равных метриках для разрешения неоднозначности к процедуре выбора минимального расстояния привлекаются значения идентификаторов коммутаторов и портов. Предпочтение отдается портам и коммутаторам с наименьшими идентификаторами. Например, у коммутатора 222 порты 1 и 2 находятся на одинаковом расстоянии до корневого коммутатора 111 — оба эти порта непосредственно связаны через сегменты *A* и *B* с коммутатором 111, а значит, получают пакеты Hello с метрикой, равной 0. Так как идентификатор порта 1 меньше идентификатора порта 2, то корневым портом коммутатора 222 выбирается порт 1.

По аналогичной причине корневым портом коммутатора 555 становится порт 1, а не порт 2. Оба эти порта получают сообщения Hello, генерируемые корневым коммутатором 111, с наименьшим значением метрики 200 000. Порт 1 получает такие сообщения по маршруту: порт 1 коммутатора 111 — сегмент *C* — порт 1 коммутатора 333 — порт 6 коммутатора 333 — сегмент *G*, соответственно порт 2 получает их по маршруту: порт 3 коммутатора 111 — сегмент *E* — порт 1 коммутатора 444 — порт 6 коммутатора 444 — сегмент *I*.

Третий этап — выбор назначенных коммутаторов и портов для каждого сегмента сети.

Назначенным является тот коммутатор (из числа коммутаторов, непосредственно подключенных к данному сегменту), у которого расстояние до корневого моста является минимальным (точнее, расстояние от корневого порта этого коммутатора до корневого коммутатора). Назначенные порты для сегментов исполняют ту же роль, что корневые порты для коммутаторов — они находятся на кратчайшем пути до корневого коммутатора.

Как и при выборе корневого порта, здесь используется распределенная процедура. Каждый коммутатор сегмента, прежде всего, исключает из рассмотрения свой корневой порт (для сегмента, к которому он подключен, всегда существует другой коммутатор, расположенный ближе к корню). Для каждого из оставшихся портов выполняется сравнение принятых по ним минимальных расстояний до корня (еще до наращивания на метрику сегмента) с расстоянием до корня корневого порта данного коммутатора. Если все принятые на этом порту расстояния оказываются больше, чем расстояние от собственного корневого порта, значит, для сегмента, к которому подключен порт, кратчайший путь к корневому коммутатору проходит через него, и он становится назначенным. Коммутатор делает все свои порты, для которых такое условие выполняется, назначенными. Когда имеется несколько портов с одинаковым кратчайшим расстоянием до корневого коммутатора, выбирается порт с наименьшим идентификатором.

В рассматриваемом примере коммутатор 111 при проверке порта 1 обнаруживает, что через этот порт принимаются пакеты с минимальным расстоянием 200 000 (это пакеты от порта 1 коммутатора 222, который ретранслирует через все свои порты сообщения Hello, полученные от коммутатора 111, но с измененной метрикой, в частности передает их и коммутатору 111). Так как коммутатор 111 является корневым, то его расстояние до корневого коммутатора равно нулю, то есть меньше, чем у получаемых через порт 1 сообщений. Поэтому коммутатор 1 объявляет свой порт 1 назначенным для сегмента *A*. Коммутатор 222 не может объявить свой порт 1 назначенным для сегмента *A*, так как через него он получает сообщения с минимальной метрикой 0, а у его корневого порта метрика равна 200 000.

На выполнение всех трех этапов коммутаторам сети отводится по умолчанию 15 с. Эта стадия работы портов называется стадией прослушивания (listening), поскольку порты слушают только сообщения BPDU и не передают пользовательских кадров. Считается, что порты находятся в заблокированном состоянии, которое относится только к пользовательским кадрам, в то время как кадры BPDU обрабатываются. Предполагается, что в стадии прослушивания каждый коммутатор получит столько пакетов Hello, сколько потребуется для определения состояния своих портов.

Все остальные порты, кроме корневых и назначенных, каждым коммутатором блокируются и не могут передавать пользовательские кадры. Математически доказано, что при таком выборе активных портов в сети исключаются петли, а оставшиеся связи образуют *покрывающее дерево* (если оно вообще может быть построено при существующих связях в сети).

Результат работы протокола STP для нашего примера показан на рис. 14.3.

На рисунке корневые порты коммутаторов отмечены символом *R*, назначенные порты закрашены, а заблокированные зачеркнуты.

После построения покрывающего дерева коммутатор начинает принимать (но не продвигать) пакеты данных и на основе их адресов источника строить таблицу продвижения. Это обычный режим обучения прозрачного моста, который ранее нельзя было активизировать, так как порт не был уверен в том, что он останется корневым или назначенным и будет передавать пакеты данных. Стадия обучения (learning) также выдерживается в течение интервала 15 с. При этом порт продолжает участвовать в работе алгоритма STA, так что поступление пакетов BPDU с лучшими параметрами переводит его в заблокированное состояние.

И только после двукратной выдержки по таймеру порт переходит в стадию продвижения (forwarding) и начинает продвигать пользовательские кадры в соответствии с построенной таблицей (которая продолжает модифицироваться, отражая изменения в структуре сети). Фактически в нашем примере в продвижении пользовательских пакетов после построения активной топологии участвуют только коммутаторы 111, 333 и с 555 по 888.

В процессе нормальной работы корневой коммутатор продолжает генерировать пакеты Hello, а остальные коммутаторы получают их через свои корневые порты и ретранслируют через назначенные порты. У коммутатора могут отсутствовать назначенные порты, как у коммутаторов 222 и 444, но он все равно участвует в работе протокола STA, так как корневой порт принимает служебные пакеты BPDU.

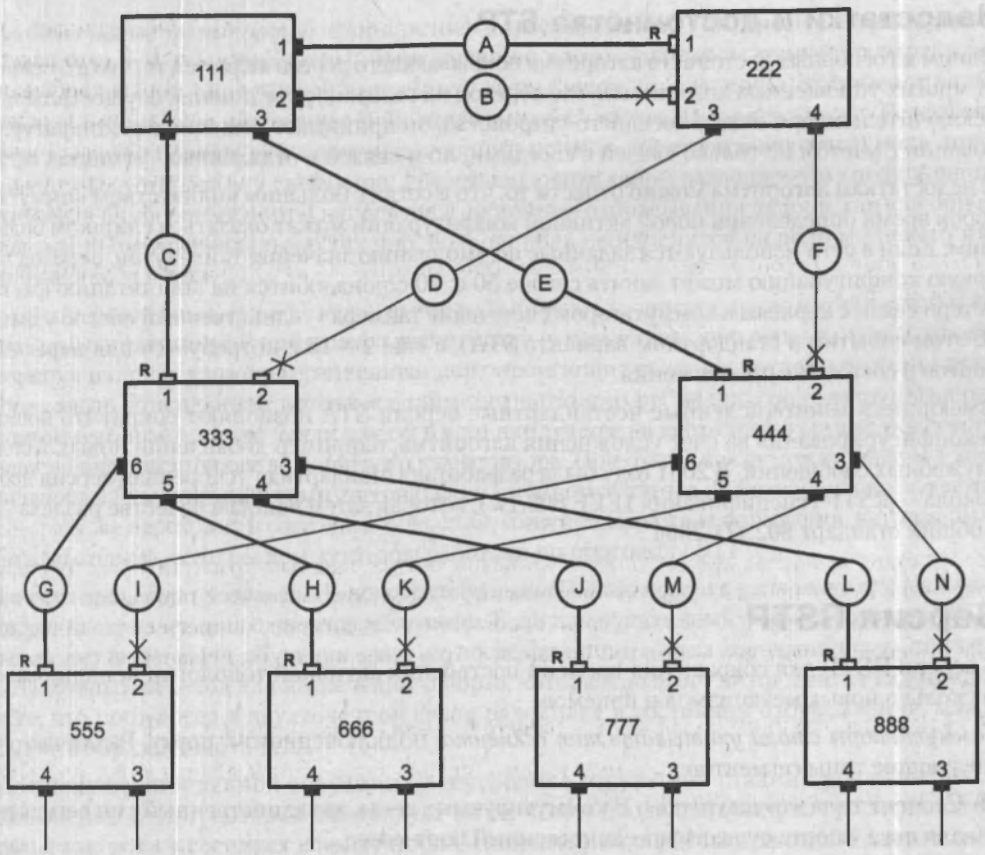


Рис. 14.3. Активная топология, найденная по протоколу STP

Если по истечении максимального времени жизни сообщения (по умолчанию — 10 интервалов Hello, то есть 20 с) корневой порт любого коммутатора сети не получает служебный пакет Hello, то он инициализирует новую процедуру построения покрывающего дерева. При этом на все порты генерируется и передается пакет Hello, в котором коммутатор указывает себя в качестве корневого. Аналогичным образом ведут себя и другие коммутаторы сети, у которых сработал таймер истечения максимального времени жизни сообщения, в результате чего выбирается новая активная конфигурация.

В процессе изменения активной топологии адресная информация, находящаяся в таблицах продвижения коммутаторов, может перестать соответствовать действительности, так как некоторые порты изменяют свое состояние с активного на заблокированное, и наоборот. Использование устаревшей адресной информации может приводить к тому, что некоторое время кадры будут посылаться в неверном направлении и не доходить до адресатов. Для того чтобы сообщить коммутатору о том, что в сети произошло изменение топологии и необходимо удалить старую адресную информацию, по сети распространяются уведомления об изменении конфигурации (это особый тип пакета BPDU).

Недостатки и достоинства STP

Одним из основных достоинств алгоритма покрывающего дерева является то, что в отличие от многих упрощенных алгоритмов, где переход на резервное соединение осуществляется исключительно при отказе соседнего устройства, он принимает решение о реконфигурировании с учетом не только связей с соседями, но и связей в отдаленных сегментах сети.

К недостаткам алгоритма можно отнести то, что в сетях с большим количеством коммутаторов время определения новой активной конфигурации может оказаться слишком большим. Если в сети используются заданные по умолчанию значения тайм-аутов, переход на новую конфигурацию может занять свыше 50 с: 20 с понадобится на констатацию факта потери связи с корневым коммутатором (истечение таймера — единственный способ узнать об этом событии в стандартном варианте STA), а еще 2 × 15 с потребуется для перехода портов в состояние продвижения.

Имеющиеся многочисленные нестандартные версии STA позволяют сократить время реконфигурирования за счет усложнения алгоритма, например добавления новых типов служебных сообщений. В 2001 году была разработана стандартная ускоренная версия протокола — RSTP (спецификация IEEE 802.1w), которая затем вошла в качестве раздела 17 в общий стандарт 802.1D-2004.

Версия RSTP

В версии RSTP для сокращения времени построения активной топологии использовано несколько новых механизмов и приемов.

Коммутаторы стали учитывать тип сегмента, подключенного к порту. Различаются следующие типы сегментов:

- Сегмент типа «точка-точка». В коммутируемых сетях это единственный тип сегмента; для него у порта существует единственный порт-сосед.
- Разделяемая среда. Стандарт RSTP по-прежнему учитывает существование разделяемой среды, так как формально ее никто не отменял, и все стандарты, включая основной стандарт Ethernet IEE 802.3, описывают работу сегмента этого типа.
- Тупиковая связь (edge port). Связь, которая соединяет порт коммутатора с конечным узлом сети; по этому сегменту нет смысла ожидать прихода сообщений протокола RSTP. Тупиковая связь конфигурируется администратором.

В случае подключения к порту тупикового сегмента этот порт не участвует в протоколе RSTP, а сразу после включения переходит в стадию продвижения кадров. Нужно заметить, что в стандарте RSTP начальное заблокированное состояние портов переименовано в состояние отбрасывания.

Для портов со связями остальных типов переход в состояние продвижения по-прежнему достижим только после нахождения в стадии обучения.

Исключается стадия прослушивания. Коммутаторы не выдерживают паузу в 15 с для того, чтобы зафиксировать соответствующую роль порта, например корневого или назначенного. Вместо этого порты переходят в стадию обучения сразу же после назначения им роли корневого или назначенного порта.

Сокращается период фиксации отказа в сети — вместо 10 периодов неполучения сообщений Hello он стал равен трем таким периодам, то есть 6 с вместо 20.

Введены новые роли портов – появились **альтернативный** (alternative) и **резервный** (backup) порты. Альтернативный порт является портом-дублером корневого порта коммутатора, то есть он начинает продвигать кадры в том случае, когда отказывает (либо перестает принимать сообщения Hello в течение трех периодов) корневой порт. Резервный порт является портом-дублером назначенного порта сегмента; однако такая роль порта имеет смысл только для сегментов, представляющих собой разделяемую среду. Альтернативные и резервные порты находятся в состоянии отбрасывания кадров, так как они не должны продвигать кадры до тех пор, пока их роль не изменится на роль корневого или назначенного порта.

Как альтернативные, так и резервные порты выбираются одновременно с корневыми и назначенными портами. Такой подход значительно ускоряет реакцию сети на отказы, так как переход, например, на альтернативный порт происходит сразу же после фиксации отказа и не связан с ожиданием истечения тайм-аутов. Например, на рис. 14.3 альтернативным портом выбирается порт 2, так как он имеет наилучшее из всех портов (после корневого, естественно) расстояние до корневого коммутатора. При отказе связи между портом 4 коммутатора 111 и портом 1 коммутатора 333 порт 2 коммутатора 333 становится корневым. Он сразу же переходит в состояние обучения, минуя стадию прослушивания, которая была бы необходима, если бы коммутаторы работали по протоколу STP.

Введена процедура подтверждения перехода назначенного порта в состояние продвижения кадров после изменения активной топологии. Если альтернативный порт в протоколе RSTP переходит в состояние обучения сразу же после фиксации отказа корневого порта, то такой безусловный переход для назначенного порта, который до этого не продвигал кадры из-за того, что порт-сосед в двухточечной связи находился в состоянии отбрасывания, может вызвать образование петель.

Для исключения данной ситуации (и в условиях отсутствия стадии прослушивания) назначенный порт, который претендует на то, чтобы продвигать кадры, просит подтвердить свою роль у соседних коммутаторов. Например, на рис. 14.3 при отказе связи между портом 4 коммутатора 111 и портом 1 коммутатора 333 порт 3 коммутатора 222 должен инициировать процедуру подтверждения.

Для этого порт посылает своему соседу по сегменту «точка-точка» конфигурационное сообщение, называемое предложением (proposing). Это сообщение вызывает временный перевод всех назначенных портов соседа в состояние отбрасывания; кроме того, эти порты распространяют сообщение с предложением далее по сети своим нижележащим (в отношении расстояния до корневого коммутатора) соседям. Когда это сообщение доходит до коммутатора, у которого все порты либо находятся в стабильном состоянии, либо являются тупиковыми, то этот коммутатор отвечает на него сообщением согласия (название этого сообщения отражает тот факт, что коммутатор согласен на то, чтобы назначенные порты, передавшие предложение на переход в состояние продвижения кадров, стали продвигать кадры). Назначенный порт, получив в ответ сообщение согласия (а оно проходит в обратном направлении, от листьев к корню), фиксирует состояние продвижения кадров. Если же назначенный порт не получает такого сообщения, то он останется в состоянии отбрасывания. Данная процедура исключает возникновение петель в активной конфигурации, к тому же она сокращает время работы протокола RSTP, так как именно благодаря ей исключается стадия прослушивания.

За счет новых механизмов и новых ролей портов протокол RSTP строит новую активную топологию существенно быстрее, чем протокол STP – за несколько секунд вместо минуты

или даже нескольких минут. Кроме того, время построения новой активной топологии по протоколу RSTP не зависит от размера сети.

На рис. 14.4 показана активная топология покрывающего дерева, найденная для предыдущего примера сети (см. рис. 14.2), но уже по протоколу RSTP. Двумя черточками отмечены альтернативные порты, находящиеся в состоянии отбрасывания, а ромбами — тупиковые порты.

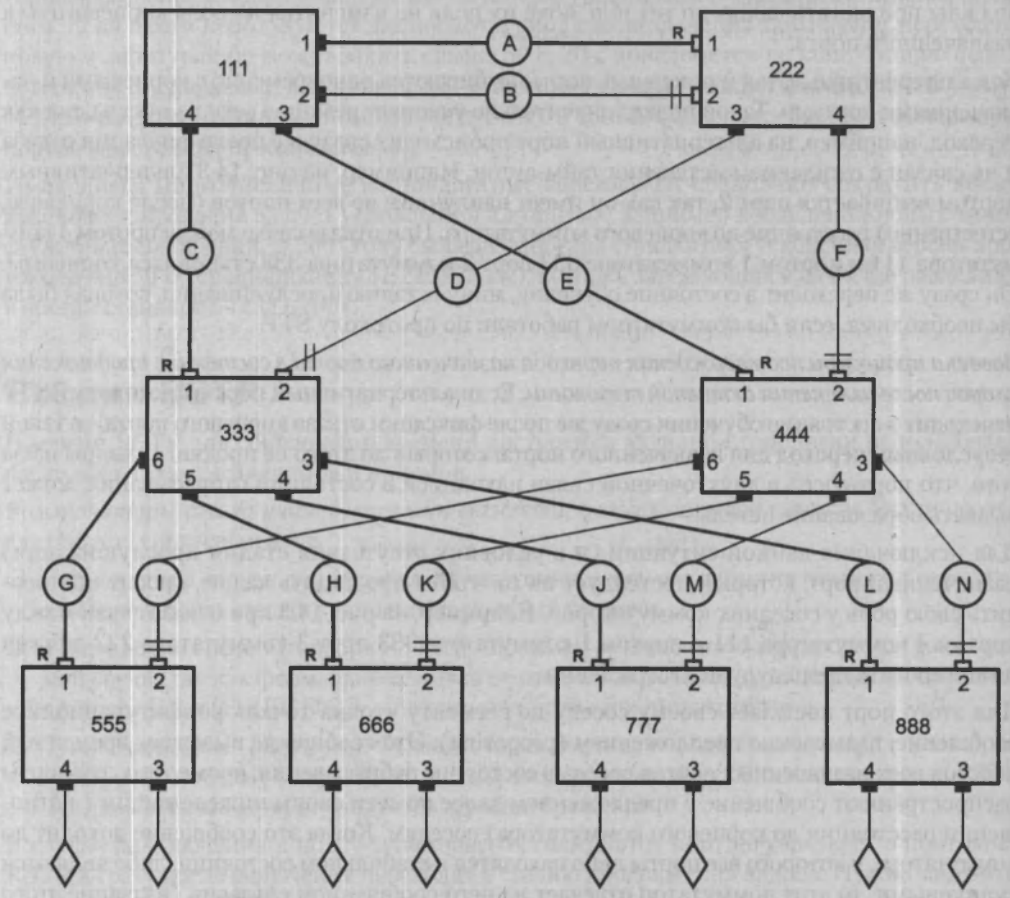


Рис. 14.4. Активная топология, найденная по протоколу RSTP

Как видно из рисунка, активная топология, найденная по обоим протоколам, совпадает. Разница только в том, что время перехода на новую топологию в случае отказа элемента сети оказывается меньше, если коммутаторы поддерживают протокол RSTP, так как альтернативные порты уже найдены и будут очень быстро использованы. Протокол RSTP совместим с протоколом STP, так что сеть, построенная из коммутаторов, часть из которых поддерживает RSTP, а часть — STP, будет работать нормально.

Агрегирование линий связи в локальных сетях

Транки и логические каналы

Агрегирование линий связи (физических каналов) между двумя коммуникационными устройствами в один логический канал является еще одной формой использования избыточных альтернативных связей в локальных сетях.

Отличие техники агрегирования линий связи от алгоритма покрывающего дерева достаточно принципиально.

- Протоколы STP и RSTP переводят избыточные связи в горячий резерв, оставляя в рабочем состоянии только минимальный набор линий, необходимых для связности сегментов сети. В этом случае *повышается надежность* сети, но не ее производительность.
- При агрегировании физических каналов все избыточные связи остаются в рабочем состоянии, в результате повышается как *надежность* сети, так и ее *производительность*.

При отказе одной из составляющих агрегированного логического канала, который часто называют **транком**, трафик распределяется между оставшимися линиями. На рис. 14.5 примером такой ситуации является транк 2, в котором один из физических каналов (центральный) отказал, так что все кадры передаются по оставшимся двум каналам. Этот пример демонстрирует повышение *надежности* при агрегировании.

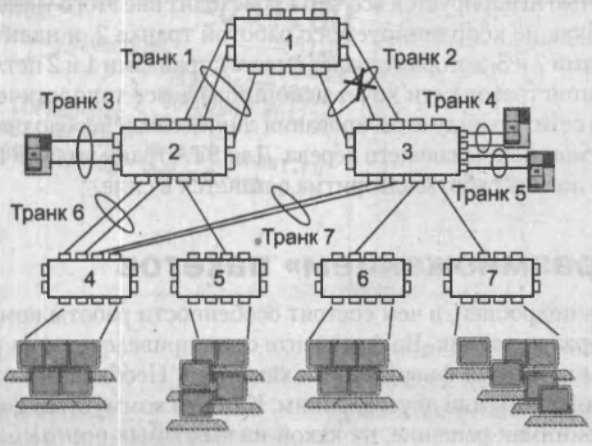


Рис. 14.5. Агрегирование физических каналов

Покажем теперь, как агрегирование линий связи повышает *производительность* сети. Так, на рисунке коммутаторы 1 и 3 соединены тремя параллельными линиями связи, что в три раза повышает производительность этого участка сети по сравнению со стандартным вариантом топологии дерева, которая не допускает таких параллельных связей. Повышение производительности связи между коммутаторами путем агрегирования линий связи в некоторых случаях является более эффективным, чем замена единственной линии связи более скоростной. Например, несмотря на то что семейство Ethernet предлагает широкий

выбор скоростей физического канала, от 10 Мбит/с до 10 Гбит/с, десятикратное повышение скорости при переходе от одного стандарта Ethernet к другому не всегда нужно и экономически оправдано. Так, если в установленных в сети коммутаторах отсутствует возможность добавления модуля с портом Gigabit Ethernet, то повышение скорости на некоторых каналах до 1000 Мбит/с потребует полной замены коммутаторов. В то же время вполне возможно, что у таких коммутаторов имеются свободные порты Fast Ethernet, поэтому скорость передачи данных можно было бы повысить, например, до 600 Мбит/с, объединив в агрегированный канал шесть портов Fast Ethernet.

Агрегирование линий связи является обобщением одного из подходов к применению альтернативных маршрутов, когда сеть заранее находит два маршрута, однако использует только один. При агрегировании отыскивается N маршрутов (где $N \geq 2$), каждый из которых используется для одного потока, а при отказе какого-либо маршрута «пострадавший» поток переводится на любой из оставшихся ($N - 1$) работающих маршрутов.

Агрегирование линий связи применяется как для связей между портами коммутаторов локальной сети, так и для связей между компьютером и коммутатором. Чаще всего этот вариант выбирают для высокоскоростных и ответственных серверов. В этом случае все сетевые адаптеры, входящие в транк, принадлежат одному компьютеру и разделяют один и тот же сетевой адрес. Поэтому для протокола IP или другого протокола сетевого уровня порты транка неразличимы, что соответствует концепции единого логического канала, лежащей в основе агрегирования.

Почти все методы агрегирования, применяемые в настоящее время, обладают существенным ограничением — в них учитываются только связи между двумя соседними коммутаторами сети и полностью игнорируется все, что происходит вне этого участка сети. Например, работа транка 1 никак не координируется с работой транка 2, и наличие обычной связи между коммутаторами 2 и 3, которая создает вместе с транками 1 и 2 петлю, не учитывается. Поэтому если администратор сети хочет использовать все топологические возможности объединения узлов сети, технику агрегирования линий связи необходимо применять *одновременно* с алгоритмом покрывающего дерева. Для STA транк должен выглядеть как одна линия связи, тогда логика работы алгоритма останется в силе.

Борьба с «размножением» пакетов

Рассмотрим теперь подробней, в чем состоят особенности работы коммутатора в случае, когда его порты образуют транк. Во фрагменте сети, приведенном на рис. 14.6, коммутаторы 1 и 2 связаны четырьмя физическими каналами. Необходимо отметить, что транк может быть односторонним или двусторонним. Каждый коммутатор контролирует только отправку кадра, принимая решение, на какой из выходных портов его нужно передать. Поэтому если оба коммутатора считают связывающие их каналы транком, то он будет двусторонним, в противном случае — односторонним.

Рисунок иллюстрирует поведение коммутатора 1 по отношению к параллельным каналам. В том случае, когда они не рассматриваются данным коммутатором как агрегированный канал, возникают проблемы с кадрами двух типов:

- кадрами с *еще не изученными* коммутатором уникальными адресами;
- кадрами, в которых указан *широковещательный* или *групповой адрес*.

Алгоритм прозрачного моста требует от коммутатора передавать кадр с неизученным (отсутствующим в таблице продвижения) адресом на все порты, кроме того, с которого кадр

был принят. При наличии параллельных каналов такой кадр будет «размножен» в количестве, равном количеству каналов — в приведенном примере коммутатор 2 примет четыре копии оригинального кадра.

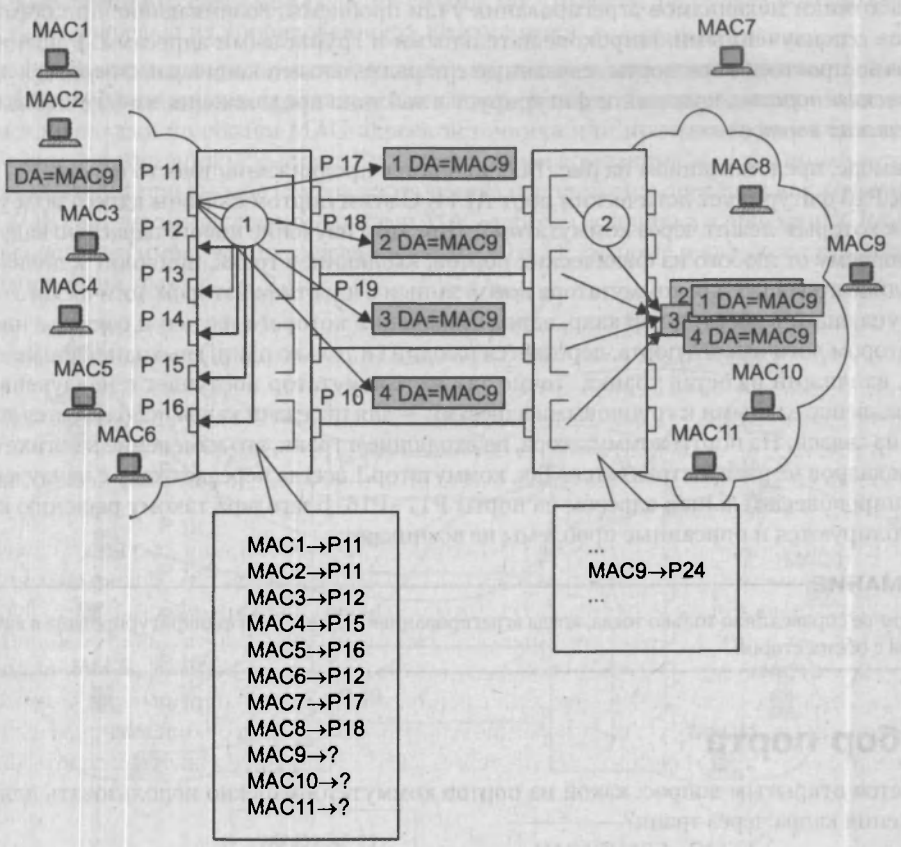


Рис. 14.6. Размножение пакетов с неизученным адресом при наличии параллельных каналов между коммутаторами

При этом происходит еще и зацикливание кадров — они будут постоянно циркулировать между двумя коммутаторами, причем удалить их из сети окажется невозможно, так как в кадрах канального уровня отсутствует поле срока жизни, часто используемое в протоколах верхних уровней, таких как IP.

В любом случае кадр с неизученным адресом повысит нагрузку на сеть за счет увеличения числа кадров, что чревато возникновением заторов, задержек и потерь данных. Помимо роста нагрузки дублирование кадров может привести к неэффективной работе многих протоколов верхнего уровня. Примером может служить узел, работающий по протоколу TCP, для которого дублирование положительных квитанций, подтверждающих факт доставки данных адресату, служит косвенным признаком перегрузки сети.

Еще больше проблем создают кадры с широковещательным адресом — они всегда должны передаваться на все порты, кроме исходного, так что в любом случае «засорение» сети посторонним трафиком окажется значительным, и кадры будут зацикливаться.

С кадрами, у которых адрес назначения изучен, проблем у коммутаторов, связанных параллельными каналами, не возникает — коммутатор передает такой кадр на тот единственный порт, по которому этот кадр впервые пришел от источника.

Разработчики механизмов агрегирования учли проблемы, возникающие при обработке кадров с неизученными, широковещательными и групповыми адресами. Решение достаточно простое — все порты, связанные с параллельными каналами, считаются одним *логическим портом*, который и фигурирует в таблице продвижения вместо нескольких *физических портов*.

В примере, представленном на рис. 14.6, в таблице продвижения вместо портов P17, P18, P19 и P10 фигурирует логический порт AL11. С этим портом связаны адреса всех узлов, путь к которым лежит через коммутатор 2. При этом изучение нового адреса по кадру, поступившему от любого из физических портов, входящих в транк, приводит к появлению в таблице продвижения коммутатора новой записи с идентификатором логического порта. Поступающий в коммутатор кадр, адрес назначения которого изучен и связан с идентификатором логического порта, передается на один (и только один!) выходной физический порт, входящий в состав транка. Точно так же коммутатор поступает с неизученными, широковещательными и групповыми адресами — для передачи кадра используется только одна из связей. На порты коммутатора, не входящие в транк, это изменение в логике обработки кадров не распространяется. Так, коммутатор 1 всегда передает кадр с неизученным или широковещательным адресом на порты P11–P16. Благодаря такому решению кадры не дублируются и описанные проблемы не возникают.

ВНИМАНИЕ

Сказанное справедливо только тогда, когда агрегированная линия связи сконфигурирована в качестве транка с обеих сторон.

Выбор порта

Остается открытым вопрос: какой из портов коммутатора нужно использовать для продвижения кадра через транк?

Можно предложить несколько вариантов ответа. Учитывая, что одной из целей агрегирования линий связи является повышение суммарной производительности участка сети между двумя коммутаторами (или коммутатором и сервером), следует распределять кадры по портам транка динамически, учитывая текущую загрузку каждого порта и направляя кадры в наименее загруженные (с меньшей длиной очереди) порты. **Динамический способ распределения кадров**, учитывающий текущую загрузку портов и обеспечивающий баланс нагрузки между всеми связями транка, должен приводить, казалось бы, к максимальной пропускной способности транка.

Однако такое утверждение справедливо не всегда, так как в нем не учитывается поведение протоколов верхнего уровня. Существует ряд таких протоколов, производительность которых может существенно снизиться, если пакеты сеанса связи между двумя конечными узлами будут приходить не в том порядке, в котором они отправлялись узлом-источником. А такая ситуация может возникнуть, если два или более последовательных кадра одного сеанса будут передаваться через разные порты транка — по причине того, что очереди в буферах этих портов имеют разную длину. Следовательно, и задержка передачи кадра может быть разной, так что более поздний кадр может обогнать более ранний.

Поэтому в большинстве реализаций механизмов агрегирования используются методы статического, а не динамического распределения кадров по портам. **Статический способ распределения кадров** подразумевает закрепление за определенным портом транка потока кадров определенного сеанса между двумя узлами, так что все кадры будут проходить через одну и ту же очередь и их упорядоченность не изменится.

Обычно при статическом распределении выбор порта для некоторого сеанса выполняется на основании определенных признаков, имеющихся в поступающих пакетах. Чаще всего такими признаками являются MAC-адреса источника или приемника или оба вместе. В популярной реализации механизма Fast EtherChannel компании Cisco для коммутаторов семейства Catalyst при выборе номера порта транка используется операция исключающего ИЛИ (XOR) над двумя последними битами MAC-адресов источника и приемника. Результат этой операции имеет четыре значения: 00, 01, 10 и 11, которые и являются условными номерами портов транка.

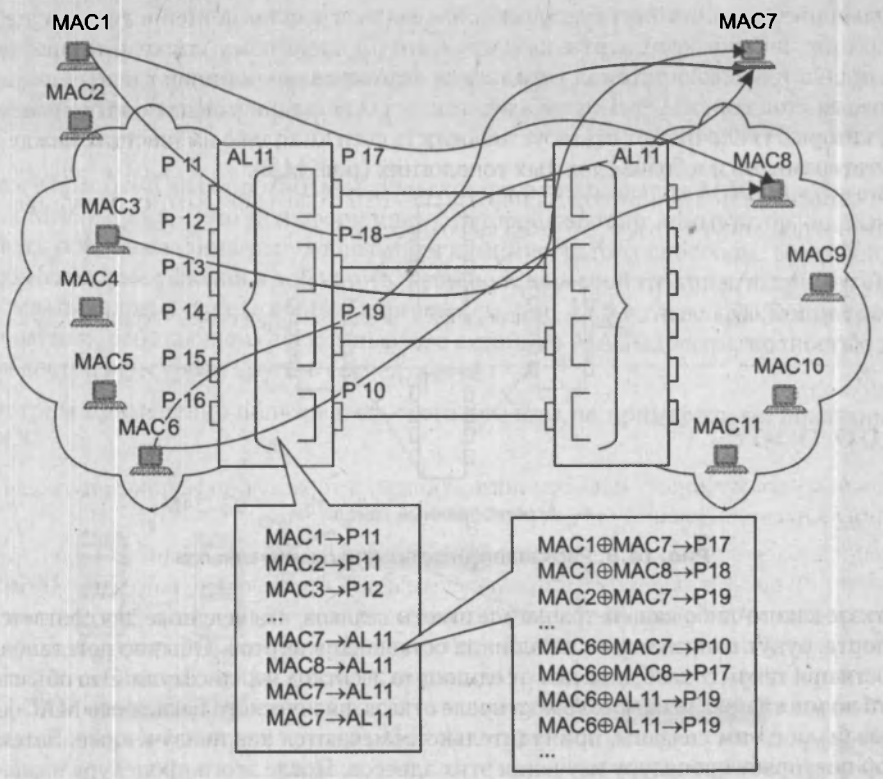


Рис. 14.7. Пример сети с механизмом Fast EtherChannel

На рис. 14.7 приведен пример сети, в которой работает механизм Fast EtherChannel. Распределение потоков для сеансов между конечными узлами получается при этом достаточно случайным. Так как в распределении не учитывается реальная нагрузка, которую создает каждый сеанс, общая пропускная способность транка может использоваться нерационально, особенно если интенсивности сеансов намного отличаются друг от друга. Кроме того, алгоритм распределения не гарантирует даже равномерного в количественном отношении

распределения сеансов по портам. Случайный набор MAC-адресов в сети может привести к тому, что через один порт будут проходить несколько десятков сеансов, а через другой — только два-три. Выравнивания нагрузки портов в данном алгоритме можно достигнуть только при большом количестве компьютеров и сеансов связи между ними.

Можно предложить и другие способы распределения сеансов по портам. Например, в соответствии с IP-адресами пакетов, которые инкапсулированы в кадры канального уровня, типами прикладных протоколов (почта по одному порту, веб-трафик по другому и т. д.). Полезным оказывается назначение порту сеансов с MAC-адресами, которые были изучены как идущие именно через этот порт — тогда трафик сеанса пойдет через один и тот же порт в обоих направлениях.

Стандартный способ создания агрегированных каналов, описанный в спецификации 802.3ad, предполагает возможность создания логического порта путем объединения нескольких физических портов, принадлежащих разным коммутаторам. Для того чтобы коммутаторы могли автоматически обеспечиваться информацией о принадлежности какого-либо физического порта определенному логическому порту, в спецификации предложен служебный **протокол управления агрегированием линий связи** (Link Control Aggregation Protocol, LCAP). Поэтому возможны такие конфигурации агрегированных каналов, которые увеличивают отказоустойчивость сети не только на участках между двумя коммутаторами, но и в более сложных топологиях (рис. 14.8).

Рис. 14.8. Распределенное агрегирование каналов

При отказе какого-либо канала транка все пакеты сеансов, назначенные для соответствующего порта, будут направляться на один из оставшихся портов. Обычно восстановление связности при таком отказе занимает от единиц до десятков миллисекунд. Это объясняется тем, что во многих реализациях транка после отказа физического канала все MAC-адреса, которые были с ним связаны, принудительно помечаются как неизученные. Затем коммутатор повторяет процедуру изучения этих адресов. После этого процедура назначения сеанса портам выполняется заново, естественно, учитываются только работающие порты. Так как тайм-ауты в сеансах протоколов локальных сетей обычно небольшие, коротким оказывается и время восстановления соединения.

Фильтрация трафика

Локальная сеть обеспечивает взаимодействие каждого узла с каждым — это очень полезное свойство, так как не требуется производить никаких специальных действий, чтобы обе-

спечить доступ узла *A* к узлу *B* — достаточно того, что эти узлы подключены к одной и той же локальной сети. В то же время в сети могут возникать ситуации, когда такая тотальная доступность узлов нежелательна. Примером может служить сервер финансового отдела, доступ к которому желательно разрешить только с компьютеров нескольких конкретных сотрудников этого отдела. Конечно, доступ можно ограничить на уровне операционной системы или системы управления базой данных самого сервера, но для надежности желательно иметь несколько эшелонов защиты и ограничить доступ еще и на уровне сетевого трафика.

Многие модели коммутаторов позволяют администраторам задавать дополнительные условия фильтрации кадров наряду со стандартными условиями их фильтрации в соответствии с информацией адресной таблицы. Такие фильтры называют пользовательскими.

Пользовательский фильтр, который также часто называют **списком доступа** (access list), предназначен для создания дополнительных барьеров на пути кадров, что позволяет ограничивать доступ определенных групп пользователей к отдельным службам сети. Пользовательский фильтр — это набор условий, которые ограничивают обычную логику передачи кадров коммутаторами.

Наиболее простыми являются пользовательские фильтры на основе MAC-адресов станций. Так как MAC-адреса — это та информация, с которой работает коммутатор, он позволяет создавать подобные фильтры удобным для администратора способом, возможно, проставляя некоторые условия в дополнительном поле адресной таблицы, например условие отбрасывать кадры с определенным адресом (см. рис. 13.6 в главе 13). Таким способом пользователю, работающему на компьютере с данным MAC-адресом, полностью запрещается доступ к ресурсам другого сегмента сети.

Рассмотрим применение пользовательского фильтра на примере сети, показанной на рис. 14.9.

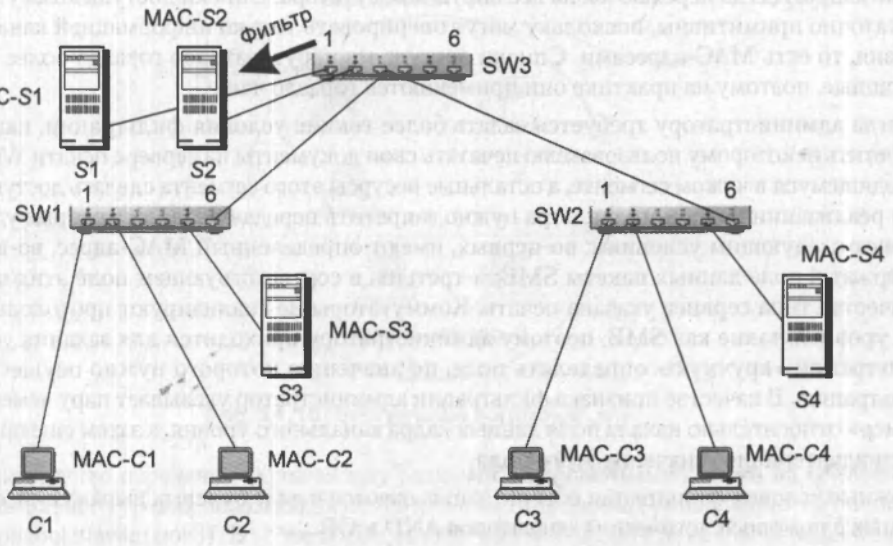


Рис. 14.9. Контроль доступа к серверу с помощью пользовательского фильтра

Пусть мы хотим разрешить доступ к серверу *S*1 только с компьютеров *C*1 и *C*3, кадры от всех остальных компьютеров до этого сервера доходить не должны. Список доступа, который решает эту задачу, может выглядеть так:

```
10 permit MAC-C1 MAC-S1
20 permit MAC-C3 MAC-S1
30 deny any any
```

Числа 10, 20 и 30 — это номера строк данного списка. Строки нумеруются с интервалом 10 для того, чтобы в дальнейшем была возможность добавить в этот список другие записи, сохраняя исходную последовательность строк. Первое условие разрешает (permit) передачу кадра, если его адрес источника равен MAC-C1, а адрес назначения — MAC-S1; второе условие делает то же, но для кадра с адресом источника MAC-C3, третье условие запрещает (deny) передачу кадров с любыми (any) адресами.

Для того чтобы список доступа начал работать, его нужно применить к трафику определенного направления на какому-либо порту коммутатора: либо к входящему, либо к исходящему. В нашем примере нужно применить список доступа к исходящему трафику порта 1 коммутатора SW3, к которому подключен сервер *S*1. Коммутатор SW3, перед тем как предать кадр на порт 1, будет просматривать условия списка доступа по очереди. Если какое-то условие из списка соблюдается, то коммутатор выполняет действие этого условия для обрабатываемого кадра, и на этом применение списка доступа для данного кадра заканчивается.

Поэтому когда от компьютера *C*1 приходит кадр, адресованный серверу *S*1, то соблюдается первое условие списка, которое разрешает передачу кадра, так что коммутатор выполняет стандартное действие по продвижению кадра, и тот доходит до сервера *S*2. С кадром от компьютера *C*3 совпадение происходит при проверке второго условия, и он также передается. Однако когда приходят кадры от других компьютеров, например компьютера *C*2, то ни первое, ни второе условия не соблюдаются, зато соблюдается третье условие, поэтому кадр не передается, а отбрасывается.

Списки доступа коммутаторов не работают с широковещательными адресами Ethernet, такие кадры всегда передаются на все порты коммутатора. Списки доступа коммутаторов достаточно примитивны, поскольку могут оперировать только информацией канального уровня, то есть MAC-адресами. Списки доступа маршрутизаторов гораздо более гибкие и мощные, поэтому на практике они применяются гораздо чаще.

Иногда администратору требуется задать более тонкие условия фильтрации, например запретить некоторому пользователю печатать свои документы на сервере печати Windows, находящемуся в чужом сегменте, а остальные ресурсы этого сегмента сделать доступными. Для реализации подобного фильтра нужно запретить передачу кадров, которые удовлетворяют следующим условиям: во-первых, имеют определенный MAC-адрес, во-вторых, содержат в поле данных пакеты SMB, в-третьих, в соответствующем поле этих пакетов в качестве типа сервиса указана печать. Коммутаторы не анализируют протоколы верхних уровней, такие как SMB, поэтому администратору приходится для задания условий фильтрации «вручную» определять поле, по значению которого нужно осуществлять фильтрацию. В качестве признака фильтрации администратор указывает пару «смещение-размер» относительно начала поля данных кадра канального уровня, а затем еще приводит шестнадцатеричное значение этого поля.

Сложные условия фильтрации обычно записываются в виде булевых выражений, формируемых с помощью логических операторов AND и OR.

Виртуальные локальные сети

Важным свойством коммутатора локальной сети является способность контролировать передачу кадров между сегментами сети. По различным причинам (соблюдение прав доступа, политика безопасности и т. д.) некоторые кадры не следует передавать по адресу назначения.

Как мы выяснили в предыдущем разделе, ограничения такого типа можно реализовать с помощью *пользовательских фильтров*. Однако пользовательский фильтр может запретить коммутатору передачу кадров только по конкретным адресам, а широковещательный трафик он *обязан* передать всем сегментам сети. Так требует алгоритм его работы. Поэтому, как уже отмечалось, сети, созданные на основе коммутаторов, иногда называют *плоскими* — из-за отсутствия барьеров на пути широковещательного трафика. Технология виртуальных локальных сетей позволяет преодолеть указанное ограничение.

Виртуальной локальной сетью (Virtual Local Area Network, VLAN) называется группа узлов сети, трафик которой, в том числе широковещательный, на канальном уровне полностью изолирован от трафика других узлов сети.

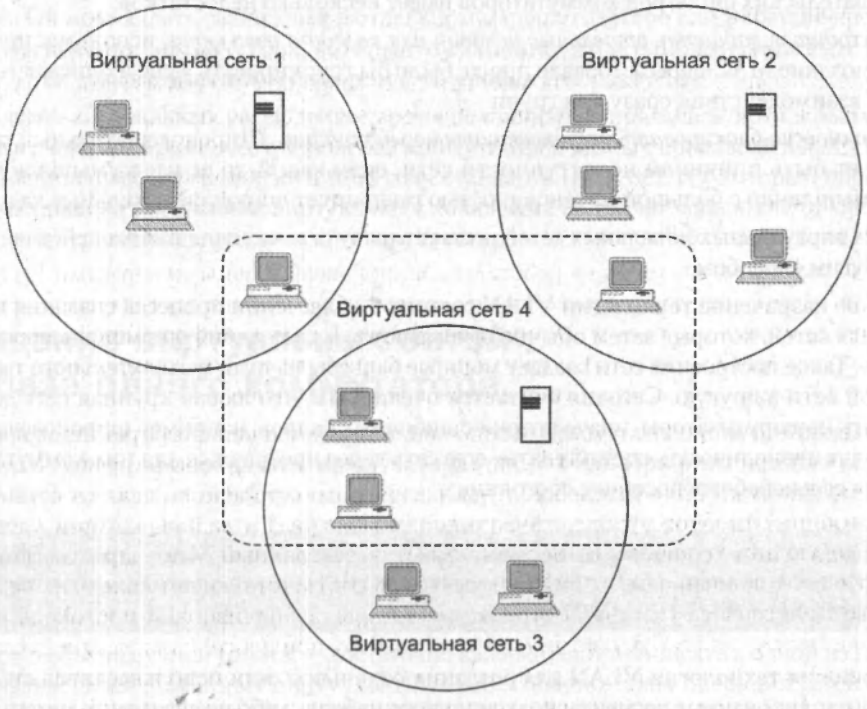


Рис. 14.10. Виртуальные локальные сети

Это означает, что передача кадров между разными виртуальными сетями на основании адреса канального уровня невозможна независимо от типа адреса (уникального, группового или широковещательного). В то же время внутри виртуальной сети кадры передаются по

технологии коммутации, то есть только на тот порт, который связан с адресом назначения кадра.

Виртуальные локальные сети могут *перекрываться*, если один или несколько компьютеров входят в состав более чем одной виртуальной сети. На рис. 14.10 сервер электронной почты входит в состав виртуальных сетей 3 и 4. Это означает, что его кадры передаются коммутаторами всем компьютерам, входящим в эти сети. Если же какой-то компьютер входит в состав только виртуальной сети 3, то его кадры до сети 4 доходить не будут, но он может взаимодействовать с компьютерами сети 4 через общий почтовый сервер. Такая схема защищает виртуальные сети друг от друга не полностью, например, широковещательный шторм, возникший на сервере электронной почты, затопит и сеть 3, и сеть 4.

Говорят, что виртуальная сеть образует *домен широковещательного трафика* по аналогии с доменом коллизий, который образуется повторителями сетей Ethernet.

Назначение виртуальных сетей

Как мы видели на примере из предыдущего раздела, с помощью пользовательских фильтров можно вмешиваться в нормальную работу коммутаторов и ограничивать взаимодействие узлов локальной сети в соответствии с требуемыми правилами доступа. Однако механизм пользовательских фильтров коммутаторов имеет несколько недостатков:

- *Приходится задавать отдельные условия для каждого узла сети*, используя при этом громоздкие MAC-адреса. Гораздо проще было бы группировать узлы и описывать условия взаимодействия сразу для групп.
- *Невозможно блокировать широковещательный трафик*. Широковещательный трафик может быть причиной недоступности сети, если какой-то ее узел умышленно или неумышленно с большой интенсивностью генерирует широковещательные кадры.

Техника виртуальных локальных сетей решает задачу ограничения взаимодействия узлов сети другим способом.

Основное назначение технологии VLAN состоит в облегчении процесса создания изолированных сетей, которые затем обычно связываются между собой с помощью маршрутизаторов. Такое построение сети создает мощные барьеры на пути нежелательного трафика из одной сети в другую. Сегодня считается очевидным, что любая крупная сеть должна включать маршрутизаторы, иначе потоки ошибочных кадров, например широковещательных, будут периодически «затапливать» всю сеть через прозрачные для них коммутаторы, приводя ее в неработоспособное состояние.

> Достоинством технологии виртуальных сетей является то, что она позволяет создавать полностью изолированные сегменты сети путем логического конфигурирования коммутаторов, не прибегая к изменению физической структуры.

До появления технологии VLAN для создания отдельной сети использовались либо физически изолированные сегменты коаксиального кабеля, либо не связанные между собой сегменты, построенные на повторителях и мостах. Затем эти сети связывались маршрутизаторами в единую составную сеть (рис. 14.11).

Изменение состава сегментов (переход пользователя в другую сеть, дробление крупных сегментов) при таком подходе подразумевает физическую перекоммутацию разъемов на

передних панелях повторителей или на кроссовых панелях, что не очень удобно в больших сетях — много физической работы, к тому же высока вероятность ошибки.

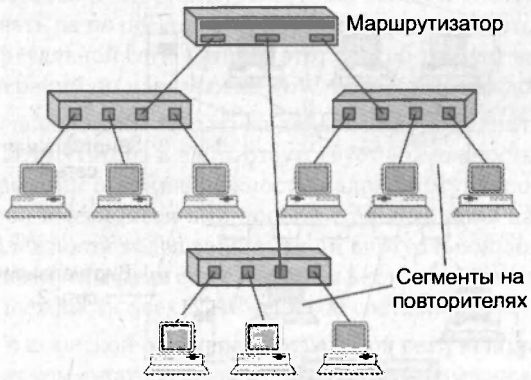


Рис. 14.11. Составная сеть, состоящая из сетей, построенных на основе повторителей

Для связывания виртуальных сетей в общую сеть требуется привлечение средств сетевого уровня. Он может быть реализован в отдельном маршрутизаторе или в составе программного обеспечения коммутатора, который тогда становится комбинированным устройством — так называемым **коммутатором 3-го уровня** (см. главу 18).

Технология виртуальных сетей долгое время не стандартизировалась, хотя и была реализована в очень широком спектре моделей коммутаторов разных производителей. Положение изменилось после принятия в 1998 году стандарта IEEE 802.1Q, который определяет базовые правила построения виртуальных локальных сетей, не зависящие от протокола канального уровня, поддерживаемого коммутатором.

Создание виртуальных сетей на базе одного коммутатора

При создании виртуальных сетей на основе одного коммутатора обычно используется механизм *группирования портов* коммутатора (рис. 14.12). При этом каждый порт приписывается той или иной виртуальной сети. Кадр, пришедший от порта, принадлежащего, например, виртуальной сети 1, никогда не будет передан порту, который не принадлежит этой виртуальной сети. Порт можно приписать нескольким виртуальным сетям, хотя на практике так делают редко — пропадает эффект полной изоляции сетей.

Создание виртуальных сетей путем группирования портов не требует от администратора большого объема ручной работы — достаточно каждый порт приписать к одной из нескольких заранее поименованных виртуальных сетей. Обычно такая операция выполняется с помощью специальной программы, прилагаемой к коммутатору.

Второй способ образования виртуальных сетей основан на *группировании MAC-адресов*. Каждый MAC-адрес, который изучен коммутатором, приписывается той или иной виртуальной сети. При существовании в сети множества узлов этот способ требует от администратора большого объема ручной работы. Однако при построении виртуальных сетей

на основе нескольких коммутаторов он оказывается более гибким, чем группирование портов.

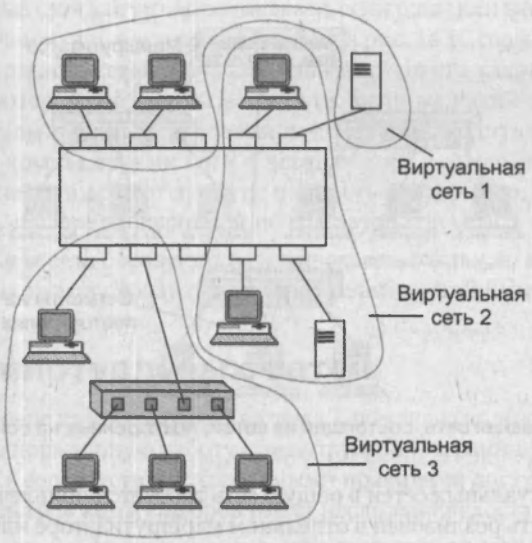


Рис. 14.12. Виртуальные сети, построенные на одном коммутаторе

Создание виртуальных сетей на базе нескольких коммутаторов

Рисунок 14.13 иллюстрирует проблему, возникающую при создании виртуальных сетей на основе нескольких коммутаторов, поддерживающих технику *группирования портов*.

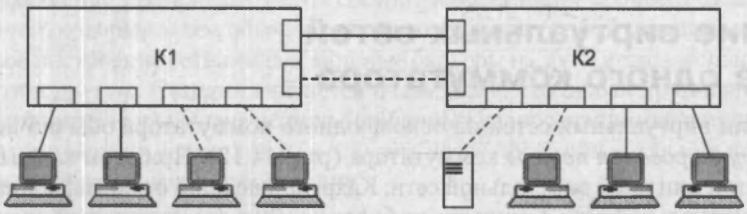


Рис. 14.13. Построение виртуальных сетей на нескольких коммутаторах с группированием портов

Если узлы какой-либо виртуальной сети подключены к разным коммутаторам, то для подключения каждой такой сети на коммутаторах должна быть выделена специальная пара портов. В противном случае, если коммутаторы будут связаны только одной парой портов, информация о принадлежности кадра той или иной виртуальной сети при передаче из коммутатора в коммутатор будет утеряна. Таким образом, коммутаторы с группированием портов требуют для своего соединения столько портов, сколько виртуальных сетей они поддерживают. Порты и кабели используются в этом случае очень расточительно. Кроме того, при соединении виртуальных сетей через маршрутизатор для каждой виртуальной

сети выделяется отдельный кабель и отдельный порт маршрутизатора, что также приводит к большим накладным расходам.

Группирование MAC-адресов в виртуальную сеть на каждом коммутаторе избавляет от необходимости связывать их по нескольким портам, поскольку в этом случае MAC-адрес становится меткой виртуальной сети. Однако этот способ требует выполнения большого количества ручных операций по маркировке MAC-адресов на каждом коммутаторе сети.

Описанные два подхода основаны только на добавлении дополнительной информации к адресным таблицам коммутатора и в них отсутствует возможность встраивания в передаваемый кадр информации о принадлежности кадра виртуальной сети. В остальных подходах используются имеющиеся или дополнительные поля кадра для сохранения информации о принадлежности кадра той или иной виртуальной локальной сети при его перемещениях между коммутаторами сети. При этом нет необходимости помнить в каждом коммутаторе о принадлежности всех MAC-адресов составной сети виртуальным сетям.

Дополнительное поле с пометкой о номере виртуальной сети используется только тогда, когда кадр передается от коммутатора к коммутатору, а при передаче кадра конечному узлу оно обычно удаляется. При этом модифицируется протокол взаимодействия «коммутатор-коммутатор», а программное и аппаратное обеспечение конечных узлов остается неизменным. До принятия стандарта IEEE 802.1Q существовало много фирменных протоколов этого типа, но все они имели один недостаток — оборудование различных производителей при образовании VLAN оказывалось несовместимым.

Этот стандарт вводит в кадре Ethernet дополнительный заголовок, который называется тегом виртуальной локальной сети.

Тег виртуальной локальной сети состоит из поля **TCI** (Tag Control Information — управляющая информация тега) размером в 2 байта и предшествующего ему поля **EtherType**, которое является стандартным для кадров Ethernet и также состоит из двух байтов (рис. 14.14).

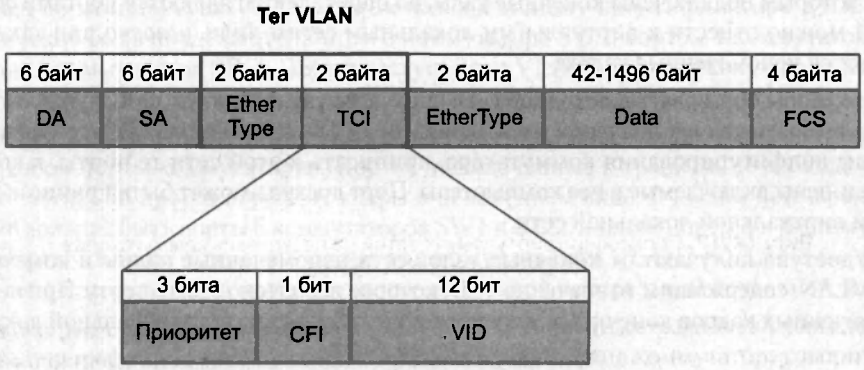


Рис. 14.14. Структура помеченного кадра Ethernet

Тег VLAN не является обязательным для кадров Ethernet. Кадр, у которого имеется такой заголовок, называют **помеченным** (tagged frame). Коммутаторы могут одновременно работать как с помеченными, так и с непомеченными кадрами. Из-за добавления тега VLAN максимальная длина поля данных уменьшилась на 4 байта.

Для того чтобы оборудование локальных сетей могло отличать и понимать помеченные кадры, для них введено специальное значение поля EtherType, равное 0x8100. Это значение говорит о том, что за ним следует поле TCI, а не стандартное поле данных. Обратите внимание, что в помеченном кадре за полями тега VLAN следует другое поле EtherType, указывающее тип протокола, данные которого переносятся полем данных кадра.

В поле TCI находится 12-битное поле номера (идентификатора) VLAN, называемого *VID*. Разрядность поля VID позволяет коммутаторам создавать до 4096 виртуальных сетей. Помимо этого в поле TCI помещено 3-битное поле *приоритета* кадра. Однобитное поле *CFI* было введено с целью поддержания специального формата кадра Token Ring, для сетей Ethernet оно должно содержать значение 0.

Пользуясь значением VID в помеченных кадрах, коммутаторы сети выполняют групповую фильтрацию трафика, разбивая сеть на виртуальные сегменты, то есть на VLAN. Для поддержки этого режима каждый порт коммутатора приписывается к одной или нескольким виртуальным локальным сетям, то есть выполняется группировка портов.

Для упрощения конфигурирования сети в стандарте 802.1Q вводятся понятия линии доступа и транка.

Линия доступа связывает порт коммутатора (называемый в этом случае **портом доступа**) с компьютером, принадлежащим некоторой виртуальной локальной сети.

Транк — это линия связи, которая соединяет между собой порты двух коммутаторов; в общем случае через транк передается трафик нескольких виртуальных сетей.

Коммутаторы, поддерживающие технику VLAN, без специального конфигурирования по умолчанию работают как стандартные коммутаторы, обеспечивая соединения всех со всеми. В сети, образованной такими коммутаторами, все конечные узлы по умолчанию относятся к условной сети VLAN1 с идентификатором VID, равным 1. Все порты этой сети, к которым подключены конечные узлы, по определению являются портами доступа. VLAN1 можно отнести к виртуальным локальным сетям лишь *условно*, так как по ней передаются непомеченные кадры.

Для того чтобы образовать в исходной сети виртуальную локальную сеть, нужно в первую очередь выбрать для нее значение идентификатора VID, отличное от 1, а затем, используя команды конфигурирования коммутатора, приписать к этой сети те порты, к которым присоединены включаемые в нее компьютеры. Порт доступа может быть приписан только к одной виртуальной локальной сети.

Порты доступа получают от конечных узлов сети непомеченные кадры и помечают их тегом VLAN, содержащим то значение VID, которое назначено этому порту. При передаче же помеченных кадров конечному узлу порт доступа удаляет тег виртуальной локальной сети.

Для более наглядного описания вернемся к рассмотренному ранее примеру сети. На рис. 14.15 показано, как решается задача избирательного доступа к серверам на основе техники VLAN.

Будем считать, что поставлена задача обеспечить доступ компьютеров *C*1 и *C*3 к серверам *S*1 и *S*3, в то время как компьютеры *C*2 и *C*4 должны иметь доступ только к серверам *S*2 и *S*4.

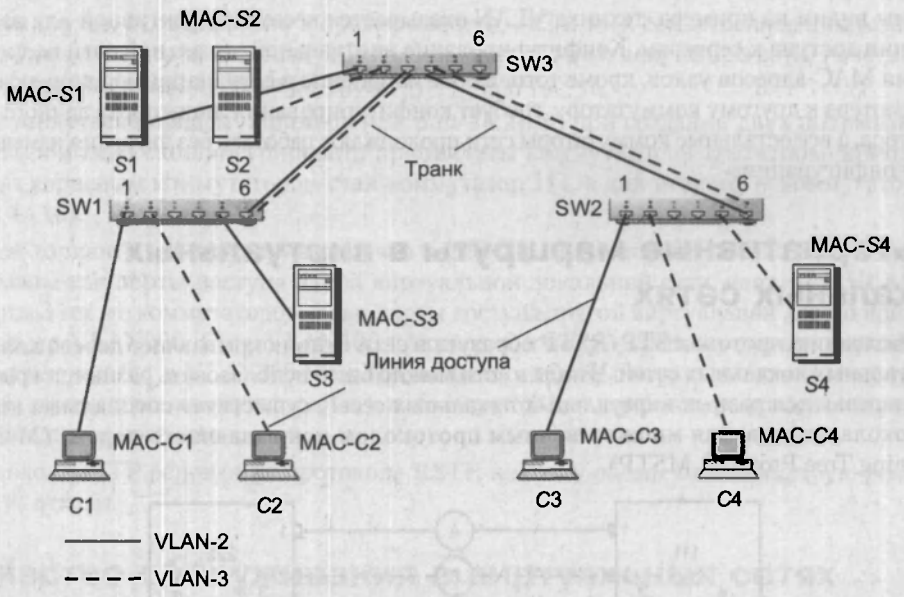


Рис. 14.15. Разбиение сети на две виртуальные локальные сети

Чтобы решить эту задачу, можно организовать в сети две виртуальные локальные сети, VLAN2 и VLAN3 (напомним, что сеть VLAN1 уже существует по умолчанию — это наша исходная сеть), приписав один набор компьютеров и серверов к VLAN2, а другой — к VLAN3.

Для приписывания конечных узлов к определенной виртуальной локальной сети соответствующие порты объявляются портами доступа этой сети путем назначения им соответствующего идентификатора VID. Например, порт 1 коммутатора SW1 должен быть объявлен портом доступа VLAN2 путем назначения ему идентификатора VID2, то же самое должно быть проделано с портом 5 коммутатора SW1, портом 1 коммутатора SW2 и портом 1 коммутатора SW3. Порты доступа сети VLAN3 должны получить идентификатор VID3.

В нашей сети нужно также организовать транки — те линии связи, которые соединяют между собой порты коммутаторов. Порты, подключенные к транкам, не добавляют и не удаляют теги, они просто передают кадры в неизменном виде. В нашем примере такими портами должны быть порты 6 коммутаторов SW1 и SW2, а также порты 3 и 4 коммутатора SW3. Порты в нашем примере должны поддерживать сети VLAN2 и VLAN3 (и VLAN1, если в сети есть узлы, явно не приписанные ни к одной виртуальной локальной сети).

Коммутаторы, поддерживающие технологию VLAN, осуществляют дополнительную фильтрацию трафика. В том случае если таблица продвижения коммутатора говорит о том, что пришедший кадр нужно передать на некоторый порт, перед передачей коммутатор проверяет, соответствует ли значение VID в теге VLAN кадра той виртуальной локальной сети, которая приписана к этому порту. В случае соответствия кадр передается, несоответствия — отбрасывается. Непомеченные кадры обрабатываются аналогичным образом, но с использованием условной сети VLAN1. MAC-адреса изучаются коммутаторами сети отдельно по каждой виртуальной локальной сети.

Как мы видим из примера, техника VLAN оказывается весьма эффективной для разграничения доступа к серверам. Конфигурирование виртуальной локальной сети не требует знания MAC-адресов узлов, кроме того, любое изменение в сети, например подключение компьютера к другому коммутатору, требует конфигурирования лишь порта данного коммутатора, а все остальные коммутаторы сети продолжают работать без внесения изменений в их конфигурации.

Альтернативные маршруты в виртуальных локальных сетях

По умолчанию протокол STP/RSTP образует в сети одно покрывающее дерево для всех виртуальных локальных сетей. Чтобы в сети можно было использовать разные покрывающие деревья для разных виртуальных локальных сетей, существует специальная версия протокола, называемая **множественным протоколом покрывающего дерева** (Multiple Spanning Tree Protocol, MSTP).

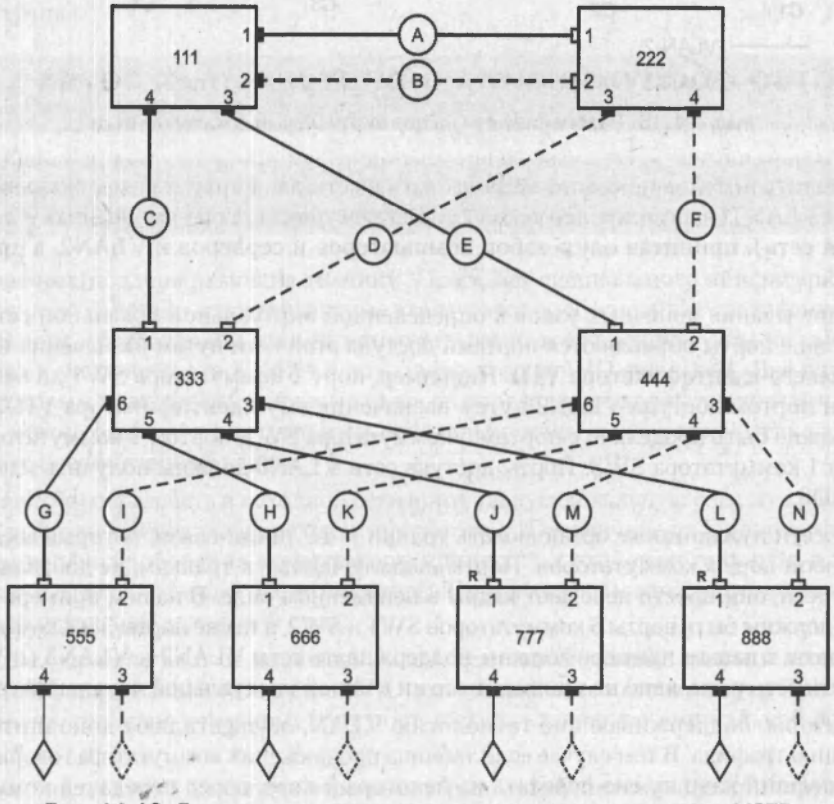


Рис. 14.16. Два покрывающих дерева, построенные по протоколу MSTP

Протокол MSTP позволяет создать несколько покрывающих деревьев и приписывать к ним различные виртуальные локальные сети. Обычно создается небольшое количество деревьев, например два или три, чтобы сбалансировать нагрузку на коммутаторы, в про-

тивном случае, как мы видели в примере на рис. 14.2 и 14.3, единственное покрывающее дерево может полностью оставить без работы некоторые коммутаторы сети, то есть недоиспользует имеющие сетевые ресурсы.

Если вернуться к нашему примеру (см. рис. 14.2), то при создании двух покрывающих деревьев можно сконфигурировать приоритеты коммутаторов так, чтобы для одного дерева корневым коммутатором стал коммутатор 111, а для второго — коммутатор 222 (рис. 14.16).

В этом варианте мы подразумеваем, что порты 4 коммутаторов с 555 по 888 сконфигурированы как порты доступа одной виртуальной локальной сети, например VLAN100, а порты 3 тех же коммутаторов — как порты доступа другой виртуальной локальной сети, например VLAN200. Сеть VLAN100 приписана к покрывающему дереву с корневым коммутатором 111, а VLAN200 — к покрывающему дереву с корневым коммутатором 222. В этом варианте все коммутаторы сети используются для передачи трафика, что повышает производительность сети.

Протокол MSTP основан на протоколе RSTP, поэтому обеспечивает быструю реакцию сети на отказы.

Качество обслуживания в виртуальных сетях

Коммутаторы локальных сетей поддерживают практически все механизмы QoS, которые мы обсуждали в главе 7. Это утверждение относится к коммутаторам локальных сетей как к классу коммуникационных устройств, каждая же конкретная модель коммутатора может быть наделена только определенным набором механизмов поддержания параметров QoS или же не иметь их вовсе. Как правило, коммутаторы рабочих групп средств QoS не поддерживают, в то время как для магистральных коммутаторов эта поддержка является обязательной.

Классификация трафика

Коммутаторы локальных сетей являются устройствами второго уровня, которые анализируют заголовки только протоколов канального уровня. Поэтому коммутаторы обычно используют для *классификации трафика* только MAC-адреса источника и приемника, а также номер порта, через который поступил кадр. Возможен также учет при классификации значения произвольного подполя внутри поля данных, заданного путем указания смещения в байтах. Эти способы не очень удобны для администратора, которому необходимо, например, отделить голосовой трафик от трафика передачи файлов. Поэтому некоторые коммутаторы, не поддерживая протоколы верхних уровней в полном объеме (например, не применяя протокол IP для продвижения пакетов), выполняют классификацию на основе признаков, содержащихся в заголовках пакетов этих протоколов — IP-адресах и портах TCP/UDP.

Маркирование трафика

Маркирование трафика обычно выполняется на границе сети, а затем его результаты используются во всех промежуточных устройствах сети. В кадре Ethernet 802.3 отсутствует поле, в которое можно было бы поместить результат маркировки трафика. Однако этот недостаток исправляет спецификация 802.1p, в которой имеются три бита дополнительного заголовка 802.1Q/p для хранения приоритета кадра.

Фактически, эти три бита служат для хранения признака одного из восьми классов трафика. Именно так трактует это поле стандарт 802.1D-2004, куда вошла спецификация 802.1p. В приложении G стандарта 802.1D-2004 даются рекомендации по разделению всего трафика локальных сетей на семь классов:

- *NC* (управление сетью). Управлению сетью дается высший приоритет при обслуживании, так как от своевременного принятия решения и доставки управляющей информации сетевым устройствам зависят любые характеристики сети.
- *VO* (голос). Голосовому трафику требуется обеспечить задержу менее 10 мс.
- *VI* (видео). Видеотрафику требуется обеспечить задержу менее 100 мс.
- *CL* (контролируемая нагрузка). При применении важных бизнес-приложений требуется некоторая форма контроля допуска (admission control) и резервирование пропускной способности для потока.
- *EE* (улучшенное обслуживание). Это улучшенный вариант обслуживания по возможности, не дающий никаких гарантий пропускной способности.
- *BE* (обслуживание по возможности, или с максимальными усилиями). Стандартное обслуживание в локальных сетях.
- *BK* (фоновый трафик). Наименее чувствительный к задержкам трафик, например трафик резервного копирования, источник которого может передавать большие объемы данных, поэтому его целесообразно выделить в особый класс, чтобы он не замедлял обработку других типов трафика.

Управление очередями

Коммутатор, поддерживающий параметры QoS, позволяет использовать несколько очередей для дифференцированной обработки классов трафика. Очереди могут обслуживаться в соответствии с алгоритмом приоритетной обработки, алгоритмом взвешенного обслуживания или на основе комбинации этих алгоритмов.

Коммутатор обычно поддерживает некоторое максимальное количество очередей, которое может оказаться меньше, чем требуемое число классов трафика. В этой ситуации несколько классов будут обслуживаться одной очередью, то есть фактически сольются в один класс. Стандарт 802.1D-2004 дает рекомендации в отношении того, какие классы трафика нужно реализовывать в сети в условиях ограниченного количества очередей в коммутаторах (табл. 16.1).

При существовании только одной очереди в сети все классы трафика обслуживаются этой очередью. На самом деле все классы обслуживаются с обычным качеством (по возможности), так как за счет управления очередями улучшить качество невозможно, хотя такие возможности, как обратная связь и резервирование полосы пропускания, для общего трафика остаются.

Две очереди дают возможность дифференцированно обслуживать группы классов трафика — менее требовательные классы BK, BE и EE в одной очереди, а более требовательные классы VO, CL, VI, NC — в другой.

Дальнейшее увеличение количества очередей позволяет более дифференцированно обслуживать трафик, вплоть до рекомендуемых семи классов. Предложенная схема является только рекомендацией, администратор сети может делить трафик на классы по своему усмотрению.

Таблица 16.1. Классы трафика и количество очередей

Количество очередей	Классы трафика
1	{BE, EE, BK, VO, CL, VI, NC}
2	{BE, EE, BK} {VO, CL, VI, NC}
3	{BE, EE, BK} {CL, VI} {VO, NC}
4	{BK} {BE, EE} {CL, VI} {VO, NC}
5	{BK} {BE, EE} {CL} {VI} {VO, NC}
6	{BK} {BE} {EE} {CL} {VI} {VO, NC}
7	{BK} {BE} {EE} {CL} {VI} {VO} {NC}

Кроме того, допускается обслуживание индивидуальных потоков трафика, но при этом каждый коммутатор должен самостоятельно выделять поток из общего трафика, так как в кадре Ethernet нет поля для переноса через сеть метки потока. В качестве признака класса трафика можно использовать номер виртуальной сети. Этот признак можно также комбинировать со значениями поля приоритета кадра, получая большое число различных классов.

Резервирование и профилирование

Коммутаторы локальных сетей поддерживают методы резервирования пропускной способности интерфейсов для классов трафика или индивидуальных потоков. Обычно коммутатор разрешает назначить классу или потоку минимальную скорость передачи данных, которая гарантируется в периоды перегрузок, а также максимальную скорость передачи данных, которая контролируется механизмом профилирования.

Для коммутаторов локальных сетей не существует стандартного протокола резервирования ресурсов. Поэтому для выполнения резервирования администратор сети должен сконфигурировать каждый коммутатор сети отдельно.

Ограничения коммутаторов

Применение коммутаторов позволяет преодолеть ограничения, свойственные сетям с разделяемой средой. Коммутируемые локальные сети могут покрывать значительные территории, плавно переходя в сети мегаполисов; они могут состоять из сегментов различной пропускной способности, образуя сети с очень высокой производительностью; они могут использовать альтернативные маршруты для повышения надежности и производительности. Однако построение сложных сетей без маршрутизаторов, а только на основе коммутаторов имеет существенные ограничения.

- Серьезные ограничения по-прежнему накладываются на топологию коммутируемой локальной сети. Требование *отсутствия петель* преодолевается с помощью техники STP/RSTP/MSTP и агрегирования каналов лишь частично. Действительно, STP не позволяет задействовать все альтернативные маршруты для передачи пользовательского трафика, а агрегирование каналов разрешает так делать только на участке сети между двумя соседними коммутаторами. Подобные ограничения не позволяют применять многие эффективные топологии, пригодные для передачи трафика.
- Логические сегменты сети, расположенные между коммутаторами, *слабо изолированы* друг от друга, а именно — не защищены от так называемых широковещательных штормов. Использование же механизма виртуальных сетей, реализованного во многих коммутаторах, хотя и позволяет достаточно гибко создавать изолированные по трафику группы станций, при этом изолирует их полностью, то есть так, что узлы одной виртуальной сети не могут взаимодействовать с узлами другой виртуальной сети.
- В сетях, построенных на основе мостов и коммутаторов, достаточно *сложно решается задача фильтрации трафика* на основе данных, содержащихся в пакете. В таких сетях фильтрация выполняется только с помощью пользовательских фильтров, для создания которых администратору приходится иметь дело с двоичным представлением содержимого пакетов.
- Реализация транспортной подсистемы только средствами физического и канального уровней приводит к *недостаточно гибкой одноуровневой системе адресации*: в качестве адреса назначения используется MAC-адрес, жестко связанный с сетевым адаптером.
- У коммутаторов *ограничены возможности по трансляции протоколов* при создании гетерогенной сети. Они не могут транслировать протоколы WAN в протоколы LAN из-за различий в системе адресации этих сетей, а также различных значений максимального размера поля данных.

Наличие серьезных ограничений у протоколов канального уровня показывает, что построение на основе средств этого уровня больших неоднородных сетей является весьма проблематичным. Естественное решение в этих случаях — привлечение средств более высокого сетевого уровня.

Пример коммутируемой сети завода можно найти на сайте www.olifer.co.uk в разделе «Коммутируемые сети».

Выводы

Для автоматического поддержания в сложных сетях резервных связей в коммутаторах реализуется алгоритм покрывающего дерева. Этот алгоритм описан в документе IEEE 802.1D и основан на периодической обмене коммутаторов специальными кадрами, с помощью которых выявляются и блокируются петлевидные связи в сети.

Протокол STA находит конфигурацию покрывающего дерева за три этапа. На первом этапе определяется корневой коммутатор, на втором — корневые порты, на третьем — назначенные порты сегментов.

Недостатком протокола STA 802.1D является сравнительно большое время установления новой активной конфигурации — около 50 с. Новый стандарт RSTP устраняет этот недостаток за счет предварительного выбора портов-дублеров для корневых и назначенных портов, а также введения некоторых других новых механизмов.

Агрегирование нескольких физических каналов в один логический является одной из форм использования нескольких активных альтернативных маршрутов в локальных сетях на коммутаторах. Агрегирование каналов повышает как производительность, так и надежность сети.

Агрегированный канал может быть образован не только между двумя соседними коммутаторами, но и распределяться между портами нескольких коммутаторов. Для автоматического уведомления о принадлежности физического порта определенному агрегированному порту разработан протокол LCAP.

Технология виртуальных локальных сетей (VLAN) позволяет в сети, построенной на коммутаторах, программным путем создать изолированные группы конечных узлов, между которыми отсутствует любой трафик, в том числе широковещательный.

Конфигурирование VLAN обычно ведется путем группирования портов или MAC-адресов.

Для построение виртуальной локальной сети на основе нескольких коммутаторов желательно помечать передаваемые кадры специальной меткой — тегом, идентифицирующем номер сети, которой принадлежит отправитель кадра. Стандартный формат тега VLAN определен в спецификации 802.1Q.

Протокол MSTP позволяет организовать в сети отдельные покрывающие деревья для виртуальных локальных сетей.

Коммутаторы LAN поддерживают многие механизмы QoS: классификацию и профилирование трафика, приоритетные и взвешенные очереди, резервирование пропускной способности.

Вопросы и задания

1. Для какой цели используется алгоритм покрывающего дерева? Варианты ответов:
   а) для автоматического построения связной топологии без петель;
   б) для защиты мостов от широковещательного шторма;
   в) для автоматического перехода на резервные связи при отказе узлов или основных линий связи сети.
2. Каждый ли коммутатор, участвующий в построении покрывающего дерева, имеет корневой порт?
3. Какой порт называется назначенным?
   а) имеющий минимальное расстояние до корневого коммутатора среди всех портов, которые подключены к данному сегменту;
   б) имеющий минимальное расстояние до корневого коммутатора среди всех портов данного коммутатора.

4. Может ли администратор влиять на выбор корневого коммутатора?
5. Каким образом коммутаторы решают, что выбор активной топологии завершен?
6. За счет каких усовершенствований протокол RSTP работает быстрее протокола STP? Варианты ответов:
   а) применение более быстрых процессоров коммутаторов;
   б) исключение тупиковых портов из процесса выбора корневых и назначенных портов;
   в) выбор портов-дублеров для корневых и назначенных портов;
   г) введение процедуры подтверждения новой роли назначенного порта.
7. Как взаимодействуют алгоритмы покрывающего дерева и агрегирования каналов?
8. В чем заключаются недостатки динамического способа выбора порта транка? Варианты ответов:
   а) неравномерная загрузка портов транка;
   б) нарушение порядка следования кадров, принадлежащих одному потоку;
   в) возможность потери кадров.
9. Преимуществами разбиения локальной сети на VLAN являются:
   а) локализация широковещательного трафика;
   б) повышение безопасности сети;
   в) улучшение управляемости сети;
   г) уменьшение объема ручного конфигурирования коммутаторов.
10. Каким образом можно объединить несколько виртуальных локальных сетей? Варианты ответов:
    а) приписать их к одному и тому же транку;
    б) сделать какой-либо конечный узел членом объединяемых сетей VLAN;
    в) объединить VLAN с помощью маршрутизатора.
11. Укажите способы образования VLAN:
    а) блокировка портов;
    б) группирование портов;
    в) группирование MAC-адресов;
    г) использование тегов стандарта IEEE 802.1Q.
12. Почему группирование портов плохо работает в сети, построенной на нескольких коммутаторах?
13. Можно ли одновременно использовать группирование портов и стандарт IEEE 802.1Q?
14. Должен ли алгоритм покрывающего дерева учитывать наличие в сети VLAN?

Часть IV

Сети TCP/IP

Прежде чем перейти к последним двум частям книги, давайте вспомним, что мы уже изучили в первых трех частях, и поговорим о том, с чем нам еще предстоит познакомиться. В части I на концептуальном уровне рассмотрено большинство проблем, которым посвящен этот учебник. Возможно, это самая сложная и важная часть книги — ведь от того, насколько хорошо заложен фундамент, зависит прочность основанных на нем знаний. Мы не раз обращались и будем обращаться к материалам части I в дальнейшем.

Части II и III посвящены конкретным технологиям передачи данных соответственно физического и канального уровней. В них существенно реже использовались абстрактные модели сети в виде графа или «облака», в котором «плавают» компьютеры. Вместо этого на первый план вышли конкретные протоколы, форматы кадров и реальное оборудование.

Что же ждет читателя в следующей части — части IV? Следуя логике, диктуемой моделью OSI, вслед за частями, в которых были изучены технологии физического и канального уровней, мы рассмотрим в части IV средства сетевого уровня, то есть средства, обеспечивающие возможность объединения множества сетей в единую сеть. Учитывая, что бесспорным лидером среди протоколов сетевого уровня является протокол IP, мы будем рассматривать вопросы построения объединенных сетей на его примере. При этом мы дадим по возможности широкую картину взаимодействия всех протоколов этого стека.

Заметим, что в предыдущих частях не раз затрагивались, а иногда и достаточно серьезно обсуждались вопросы межсетевого взаимодействия TCP/IP. Так, в главе 2 мы уже рассмотрели, хотя и в самом общем виде, понятие маршрутизации. В главе 4 в разделе «Модель OSI», изучая сетевой уровень, мы познакомились с понятием «составная сеть», которую можно представить как совокупность нескольких сетей (подсетей). Подсети в составной сети, которые могут быть как локальными, так и глобальными, соединяются между собой маршрутизаторами. В пределах каждой подсети все узлы взаимодействуют по единой для них технологии, например Ethernet, Token Ring, FDDI, Frame Relay, ATM. Однако ни одна из этих технологий не способна построить информационную связь между произвольно выбранными узлами, принадлежащими разным сетям. Именно эту задачу — организацию взаимодействия между любой произвольной парой узлов в «большой» составной сети — эффективно решают протоколы стека TCP/IP. В главе 5 было дано описание структуры Интернета — самой известной и масштабной сети, построенной на основе технологии TCP/IP. Читателю настоятельно рекомендуется еще раз внимательно просмотреть этот материал.

Забегая вперед, мы хотим предупредить читателя, что в последней части книги, посвященной технологиям WAN, мы еще не раз вернемся к протоколам TCP/IP. Мы рассмотрим особенности работы протокола IP «поверх» сетей ATM/FR, тесно связанную с IP технологию MPLS, а также защищенную версию протокола IP — протокол IPSec.

- Глава 15. Адресация в стеке протоколов TCP/IP
- Глава 16. Протокол межсетевого взаимодействия
- Глава 17. Базовые протоколы TCP/IP
- Глава 18. Дополнительные функции маршрутизаторов IP-сетей

ГЛАВА 15 Адресация в стеке протоколов TCP/IP

Приступая к изучению технологии TCP/IP, мы прежде всего рассмотрим структуру стека протоколов этой технологии, узнаем, как распределены функции между протоколами разных уровней, а также обсудим более общую тему уникальности стека протоколов TCP/IP, позволяющей ему доминировать в сетевом мире.

Важную часть технологии TCP/IP составляют задачи адресации, к числу которых относятся следующие:

- *Согласованное использование адресов различного типа*. Эта задача включает отображение адресов разных типов, например преобразование сетевого IP-адреса в локальный, доменного имени — в IP-адрес.
- *Обеспечение уникальности адресов*. В зависимости от типа адреса требуется обеспечивать однозначность адресации в пределах компьютера, подсети, корпоративной сети или Интернета.
- *Конфигурирование сетевых интерфейсов и сетевых приложений*.

Каждая из перечисленных задач имеет достаточно простое решение для сети, число узлов которой не превосходит нескольких десятков. Например, для отображения символьного доменного имени на IP-адрес достаточно поддерживать на каждом хосте таблицу всех символьных имен, используемых в сети, и соответствующих им IP-адресов. Столь же просто «вручную» присвоить всем интерфейсам в небольшой сети уникальные адреса. Однако в крупных сетях эти же задачи усложняются настолько, что требуют принципиально других решений.

Ключевым словом, которое характеризует подход к решению этих проблем, принятый в TCP/IP, является **масштабируемость**.

Процедуры, предлагаемые TCP/IP для назначения, отображения и конфигурирования адресов, одинаково хорошо работают в сетях разного масштаба. В этой главе наряду с собственно схемой образования IP-адресов мы познакомимся с наиболее популярными масштабируемыми средствами поддержки адресации в сетях TCP/IP: технологией бесклассовой междоменной маршрутизации, системой доменных имен, протоколом динамического конфигурирования хостов.

Стек протоколов TCP/IP

Сегодня стек TCP/IP широко используется как в глобальных, так и в локальных сетях. Этот стек имеет иерархическую структуру, в которой определено 4 уровня (рис. 15.1).

Прикладной уровень	FTP, Telnet, HTTP, SMTP, SNMP, TFTP
Транспортный уровень	TCP, UDP
Сетевой уровень	IP, ICMP, RIP, OSPF
Уровень сетевых интерфейсов	Не регламентируется

Рис. 15.1. Иерархическая структура стека TCP/IP

Прикладной уровень стека TCP/IP соответствует трем верхним уровням модели OSI: прикладному, представления и сеансовому. Он объединяет сервисы, предоставляемые системой пользовательским приложениям. За долгие годы применения в сетях различных стран и организаций стек TCP/IP накопил большое количество протоколов и служб прикладного уровня. К ним относятся такие распространенные протоколы, как протокол передачи файлов (File Transfer Protocol, FTP), протокол эмуляции терминала telnet, простой протокол передачи почты (Simple Mail Transfer Protocol, SMTP), протокол передачи гипертекста (Hypertext Transfer Protocol, HTTP) и многие другие. Протоколы прикладного уровня развертываются на хостах[1].

Транспортный уровень стека TCP/IP может предоставлять вышележащему уровню два типа сервиса:

- гарантированную доставку обеспечивает протокол управления передачей (Transmission Control Protocol, TCP);
- доставку по возможности, или с максимальными усилиями, обеспечивает протокол пользовательских дейтаграмм (User Datagram Protocol, UDP).

Для того чтобы обеспечить надежную доставку данных, протокол TCP предусматривает установление логического соединения, что позволяет ему нумеровать пакеты, подтверждать их прием квитанциями, в случае потери организовывать повторные передачи, распознавать и уничтожать дубликаты, доставлять прикладному уровню пакеты в том порядке, в котором они были отправлены. Благодаря этому протоколу объекты на хосте-отправителе и хосте-получателе могут поддерживать обмен данными в дуплексном режиме. TCP дает

[1] В Интернете (а значит, и в стеке протоколов TCP/IP) конечный узел традиционно называют *хостом*, а маршрутизатор — *шлюзом*. Далее мы будем использовать пары терминов «конечный узел» — «хост» и «маршрутизатор» — «шлюз» как синонимы, чтобы отдать дань уважения традиционной терминологии Интернета и в то же время не отказываться от современных терминов.

возможность без ошибок доставить сформированный на одном из компьютеров поток байтов на любой другой компьютер, входящий в составную сеть.

Второй протокол этого уровня, UDP, является простейшим дейтаграммным протоколом, который используется тогда, когда задача надежного обмена данными либо вообще не ставится, либо решается средствами более высокого уровня — прикладным уровнем или пользовательскими приложениями.

В функции протоколов TCP и UDP входит также исполнение роли связующего звена между прилегающими к транспортному уровню прикладным и сетевым уровнями. От прикладного протокола транспортный уровень принимает задание на передачу данных с тем или иным качеством прикладному уровню-получателю. Нижележащий сетевой уровень протоколы TCP и UDP рассматривают как своего рода инструмент, не очень надежный, но способный перемещать пакет в свободном и рискованном путешествии по составной сети.

Программные модули, реализующие протоколы TCP и UDP, подобно модулям протоколов прикладного уровня, устанавливаются на хостах.

Сетевой уровень, называемый также **уровнем Интернета**, является стержнем всей архитектуры TCP/IP. Именно этот уровень, функции которого соответствуют сетевому уровню модели OSI, обеспечивает перемещение пакетов в пределах составной сети, образованной объединением нескольких подсетей. Протоколы сетевого уровня поддерживают интерфейс с вышележащим транспортным уровнем, получая от него запросы на передачу данных по составной сети, а также с нижележащим уровнем сетевых интерфейсов, о функциях которого мы расскажем далее.

Основным протоколом сетевого уровня является межсетевой протокол (Internet Protocol, IP). В его задачу входит продвижение пакета между сетями — от одного маршрутизатора к другому до тех пор, пока пакет не попадет в сеть назначения. В отличие от протоколов прикладного и транспортного уровней, протокол IP развертывается не только на хостах, но и на всех маршрутизаторах (шлюзах). Протокол IP — это дейтаграммный протокол, работающий без установления соединений по принципу доставки с максимальными усилиями. Такой тип сетевого сервиса называют также «ненадежным».

К сетевому уровню TCP/IP часто относят протоколы, выполняющие вспомогательные функции по отношению к IP. Это, прежде всего, протоколы маршрутизации RIP и OSPF, предназначенные для изучения топологии сети, определения маршрутов и составления таблиц маршрутизации, на основании которых протокол IP перемещает пакеты в нужном направлении. По этой же причине к сетевому уровню могут быть отнесены протокол межсетевых управляющих сообщений (Internet Control Message Protocol, ICMP), предназначенный для передачи маршрутизатором источнику сведений об ошибках, возникших при передаче пакета, и некоторые другие протоколы.

Идеологическим отличием архитектуры стека TCP/IP от многоуровневой архитектуры других стеков является интерпретация функций самого нижнего уровня — **уровня сетевых интерфейсов**.

Напомним, что нижние уровни модели OSI (канальный и физический) реализуют множество функций доступа к среде передачи, формированию кадров, согласованию величин электрических сигналов, кодированию и синхронизации, а также некоторые другие. Все эти весьма конкретные функции составляют суть таких протоколов обмена данными, как Ethernet, PPP и многих других.

У нижнего уровня стека TCP/IP задача существенно проще — он отвечает только за организацию взаимодействия с подсетями разных технологий, входящими в составную сеть. TCP/IP рассматривает любую подсеть, входящую в составную сеть, как средство транспортировки пакетов между двумя соседними маршрутизаторами.

Задачу организации интерфейса между технологией TCP/IP и любой другой технологией промежуточной сети упрощенно можно свести к двум задачам:

- упаковка (инкапсуляция) IP-пакета в единицу передаваемых данных промежуточной сети;
- преобразование сетевых адресов в адреса технологии данной промежуточной сети.

Такой гибкий подход упрощает решение проблемы расширения набора поддерживаемых технологий. При появлении новой популярной технологии она быстро включается в стек TCP/IP путем разработки соответствующего стандарта, определяющего метод инкапсуляции IP-пакетов в ее кадры (например, спецификация RFC 1577, определяющая работу протокола IP через сети ATM, появилась в 1994 году вскоре после принятия основных стандартов ATM). Так как для каждой вновь появляющейся технологии разрабатываются собственные интерфейсные средства, функции этого уровня нельзя определить раз и навсегда, и именно поэтому нижний уровень стека TCP/IP не регламентируется.

Каждый коммуникационный протокол оперирует некоторой единицей передаваемых данных. Названия этих единиц иногда закрепляются стандартом, а чаще просто определяются традицией. В стеке TCP/IP за многие годы его существования образовалась устоявшаяся терминология в этой области (рис. 15.2).

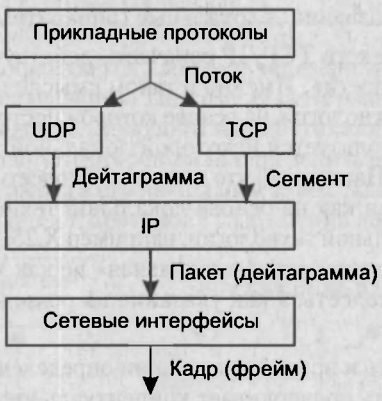


Рис. 15.2. Названия протокольных единиц данных в TCP/IP

Потоком данных, информационным потоком, или просто **потоком**, называют данные, поступающие от приложений на вход протоколов транспортного уровня — TCP и UDP.

Протокол TCP «нарезает» из потока данных **сегменты**.

Единицу данных протокола UDP часто называют **дейтаграммой**, или **датаграммой**. Дейтаграмма — это общее название для единиц данных, которыми оперируют протоколы без установления соединений. К таким протоколам относится и протокол IP, поэтому его единицу данных иногда тоже называют дейтаграммой, хотя достаточно часто используется и другой термин — **пакет**.

В стеке TCP/IP единицы данных любых технологий, в которые упаковываются IP-пакеты для их последующей передачи через сети составной сети, принято называть также **кадрами**, или **фреймам**. При этом не имеет значения, какое название используется для этой единицы данных в технологии составляющей сети. Для TCP/IP фреймом является и кадр Ethernet, и ячейка ATM, и пакет X.25 в тех случаях, когда они выступают в качестве контейнера, в котором IP-пакет переносится через составную сеть.

Типы адресов стека TCP/IP

Итак, для идентификации сетевых интерфейсов используются три типа адресов:

- локальные (аппаратные) адреса;
- сетевые адреса (IP-адреса);
- символьные (доменные) имена.

Локальные адреса

В большинстве технологий LAN (Ethernet, FDDI, Token Ring) для однозначной адресации интерфейсов используются **MAC-адреса**. Существует немало технологий (X.25, ATM, frame relay), в которых применяются другие схемы адресации. Роль, которую играют эти адреса в TCP/IP, не зависит от того, какая именно технология используется в подсети, поэтому они имеют общее название — **локальные (аппаратные) адреса**.

Слово «локальный» в контексте TCP/IP означает «действующий не во всей составной сети, а лишь в пределах подсети». Именно в таком смысле понимаются здесь термины: «локальная технология» (технология, на основе которой построена подсеть) и «локальный адрес» (адрес, который используется некоторой локальной технологией для адресации узлов в пределах подсети). Напомним, что в качестве подсети («локальной сети») может выступать сеть, построенная как на основе локальной технологии, например Ethernet, FDDI, так и на основе глобальной технологии, например X.25, Frame Relay. Следовательно, говоря о подсети, мы используем слово «локальная» не как характеристику технологии, на которой построена эта подсеть, а как указание на роль, которую играет эта подсеть в архитектуре составной сети.

Сложности могут возникнуть и при интерпретации определения «аппаратный». В данном случае термин «аппаратный» подчеркивает концептуальное представление разработчиков стека TCP/IP о подсети как о некотором вспомогательном *аппаратном* средстве, единственной функцией которого является перемещение IP-пакета через подсеть до ближайшего шлюза (маршрутизатора). И не важно, что реально нижележащая локальная технология может быть достаточно сложной, все ее сложности технологией TCP/IP игнорируются.

Рассмотрим, например, случай, когда в составную сеть TCP/IP входит сеть IPX/SPX. Последняя сама может быть разделена на подсети, и так же как IP-сеть, она идентифицирует свои узлы аппаратными и сетевыми IPX-адресами. Но технология TCP/IP игнорирует многоуровневое строение сети IPX/SPX и рассматривает в качестве локальных адресов узлов подсети IPX/SPX адреса сетевого уровня данной технологии (IPX-адреса). Аналогично, если в составную сеть включена сеть X.25, то локальными адресами узлов этой сети для протокола IP будут соответственно адреса X.25.

Сетевые IP-адреса

Чтобы технология TCP/IP могла решать свою задачу объединения сетей, ей необходима собственная глобальная система адресации, *не зависящая от способов адресации узлов в отдельных сетях*. Эта система адресации должна позволять универсальным и однозначным способом идентифицировать любой интерфейс составной сети. Очевидным решением является уникальная нумерация всех сетей составной сети, а затем нумерация всех узлов в пределах каждой из этих сетей. Пара, состоящая из **номера сети** и **номера узла**, отвечает поставленным условиям и может являться **сетевым адресом**.

В качестве номера узла может выступать либо локальный адрес этого узла (такая схема принята в стеке IPX/SPX), либо некоторое число, никак не связанное с локальной технологией и однозначно идентифицирующее узел в пределах данной подсети. В первом случае сетевой адрес становится зависимым от локальных технологий, что ограничивает его применение. Например, сетевые адреса IPX/SPX рассчитаны на работу в составных сетях, объединяющих сети, в которых используются только MAC-адреса или адреса аналогичного формата. Второй подход более универсален, он характерен для стека TCP/IP[1].

В технологии TCP/IP сетевой адрес называют **IP-адресом**.

ВНИМАНИЕ

Если рассматривать IP-сеть, то можно отметить, что маршрутизатор по определению входит сразу в несколько сетей, следовательно, каждый его интерфейс имеет собственный IP-адрес. Конечный узел также может входить в несколько IP-сетей. В этом случае компьютер должен иметь несколько IP-адресов — по числу сетевых связей. Таким образом, IP-адрес идентифицирует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.

Каждый раз, когда пакет направляется адресату через составную сеть, в его заголовке указывается IP-адрес узла назначения. По номеру сети назначения каждый очередной маршрутизатор находит IP-адрес следующего маршрутизатора. Перед тем как отправить пакет в следующую сеть, маршрутизатор должен определить на основании найденного IP-адреса следующего маршрутизатора его локальный адрес. Для этой цели протокол IP, как показано на рис. 15.3, обращается к протоколу разрешения адресов (ARP).

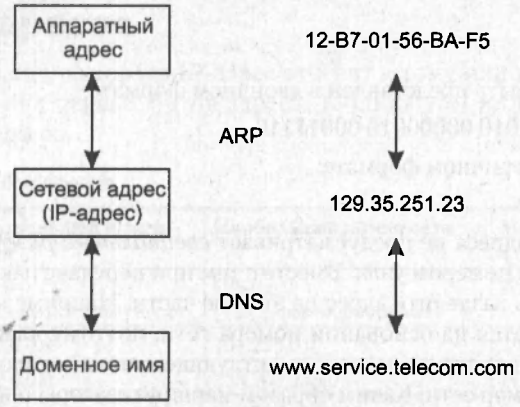


Рис. 15.3. Преобразование адресов

[1] Заметим, что использование локального адреса в качестве номера узла имеет ряд преимуществ. Как будет показано далее, именно такая схема принята в протоколе IPv6.

Доменные имена

Для идентификации компьютеров аппаратное и программное обеспечение в сетях TCP/IP полагается на IP-адреса. Например, команда ftp://192.45.66.17 будет устанавливать сеанс связи с нужным ftp-сервером, а команда http://203.23.106.33 откроет начальную страницу на корпоративном веб-сервере. Однако пользователи обычно предпочитают работать с более удобными **символьными именами** компьютеров.

Символьные идентификаторы сетевых интерфейсов в пределах составной сети строятся по иерархическому принципу. Составляющие полного символьного (или доменного) имени в IP-сетях разделяются точкой и перечисляются в следующем порядке: сначала простое имя хоста, затем имя группы хостов (например, имя организации), потом имя более крупной группы (домена) и так до имени домена самого высокого уровня (например, домена объединяющего организации по географическому принципу: RU — Россия, UK — Великобритания, US — США). Примером доменного имени может служить имя base2.sales.zil.ru.

Между **доменным именем** и IP-адресом узла нет никакой функциональной зависимости, поэтому единственный способ установления соответствия — это таблица. В сетях TCP/IP используется специальная **система доменных имен** (Domain Name System, DNS), которая устанавливает это соответствие на основании создаваемых администраторами сети таблиц соответствия. Поэтому доменные имена называют также **DNS-именами**.

В общем случае сетевой интерфейс может иметь несколько локальных адресов, сетевых адресов и доменных имен.

Формат IP-адреса

В заголовке IP-пакета для хранения IP-адресов отправителя и получателя отводятся два поля, каждое имеет фиксированную длину 4 байта (32 бита). IP-адрес состоит из двух логических частей — **номера сети** и **номера узла в сети**.

Наиболее распространенной формой представления IP-адреса является запись в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками, например:

128.10.2.30

Этот же адрес может быть представлен в двоичном формате:

10000000 00001010 00000010 00011110

А также в шестнадцатеричном формате:

80.0A.02.1D

Заметим, что запись адреса не предусматривает *специального разграничительного знака* между номером сети и номером узла. Вместе с тем при передаче пакета по сети часто возникает необходимость разделить адрес на эти две части. Например, маршрутизация, как правило, осуществляется на основании номера сети, поэтому каждый маршрутизатор, получая пакет, должен прочитать из соответствующего поля заголовка адрес назначения и выделить из него номер сети. Каким образом маршрутизаторы определяют, какая часть из 32 бит, отведенных под IP-адрес, относится к номеру сети, а какая — к номеру узла?

Можно предложить несколько вариантов решения этой проблемы.

- Простейший из них состоит в использовании **фиксированной границы**. При этом все 32-битное поле адреса заранее делится на две части не обязательно равной, но фиксированной длины, в одной из которых всегда будет размещаться номер сети, в другой — номер узла. Решение очень простое, но хорошее ли? Поскольку поле, которое отводится для хранения номера узла, имеет фиксированную длину, все сети будут иметь одинаковое максимальное число узлов. Если, например, под номер сети отвести один первый байт, то все адресное пространство распадется на сравнительно небольшое (2^8) число сетей огромного размера (2^{24} узлов). Если границу передвинуть дальше вправо, то сетей станет больше, но все равно все они будут одинакового размера. Очевидно, что такой жесткий подход не позволяет дифференцированно удовлетворять потребности отдельных предприятий и организаций. Именно поэтому он не нашел применения, хотя и использовался на начальном этапе существования технологии TCP/IP (RFC 760).
- Второй подход (RFC 950, RFC 1518) основан на использовании *маски*, которая позволяет максимально гибко устанавливать границу между номером сети и номером узла. При таком подходе адресное пространство можно использовать для создания множества сетей разного размера.

  Маска — это число, применяемое в паре с IP-адресом, причем двоичная запись маски содержит непрерывную последовательность единиц в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Граница между последовательностями единиц и нулей в маске соответствует границе между номером сети и номером узла в IP-адресе.
- И, наконец, способ, основанный на **классах адресов** (RFC 791). Этот способ представляет собой компромисс по отношению к двум предыдущим: размеры сетей хотя и не могут быть произвольными, как при использовании масок, но и не должны быть одинаковыми, как при установлении фиксированных границ. Вводится пять классов адресов: A, B, C, D, E. Три из них — A, B и C — предназначены для адресации сетей, а два — D и E — имеют специальное назначение. Для каждого класса сетевых адресов определено собственное положение границы между номером сети и номером узла.

Классы IP-адресов

Признаком, на основании которого IP-адрес относят к тому или иному классу, являются значения нескольких первых битов адреса. Таблица 15.1 иллюстрирует структуру IP-адресов разных классов.

Таблица 15.1. Классы IP-адресов

Класс	Первые биты	Наименьший номер сети	Наибольший номер сети	Максимальное число узлов в сети
A	0	1.0.0.0 (0 — не используется)	126.0.0.0 (127 — зарезервирован)	2^{24}, поле 3 байта
B	10	128.0.0.0	191.255.0.0	2^{16}, поле 2 байта
C	110	192.0.0.0	223.255.255.0	2^8, поле 1 байт
D	1110	224.0.0.0	239.255.255.255	Групповые адреса
E	11110	240.0.0.0	247.255.255.255	Зарезервировано

- К **классу A** относится адрес, в котором старший бит имеет значение 0. В адресах класса A под идентификатор сети отводится 1 байт, а остальные 3 байта интерпретируются как номер узла в сети. Сети, все IP-адреса которых имеют значение первого байта в диапазоне от 1 (00000001) до 126 (01111110), называются сетями класса A. Значение 0 (00000000) первого байта не используется, а значение 127 (01111111) зарезервировано для специальных целей (см. далее). Сетей класса A сравнительно немного, зато количество узлов в них может достигать 224, то есть 16 777 216 узлов.
- К **классу B** относятся все адреса, старшие два бита которых имеют значение 10. В адресах класса B под номер сети и под номер узла отводится по 2 байта. Сети, значения первых двух байтов адресов которых находятся в диапазоне от 128.0 (10000000 00000000) до 191.255 (10111111 11111111), называются сетями класса B. Ясно, что сетей класса B больше, чем сетей класса A, а размеры их меньше. Максимальное количество узлов в сетях класса B составляет 2^{16} (65 536).
- К **классу C** относятся все адреса, старшие три бита которых имеют значение 110. В адресах класса C под номер сети отводится 3 байта, а под номер узла — 1 байт. Сети, старшие три байта которых находятся в диапазоне от 192.0.0 (11000000 00000000 00000000) до 223.255.255 (11011111 11111111 11111111), называются сетями класса C. Сети класса C наиболее распространены, и наименьшее максимальное число узлов в них равно 2^8 (256).
- Если адрес начинается с последовательности 1110, то он является адресом **класса D** и обозначает особый **групповой адрес** (multicast address). В то время как адреса классов A, B и C служат для идентификации отдельных сетевых интерфейсов, то есть являются **индивидуальными адресами** (unicast address), групповой адрес идентифицирует группу сетевых интерфейсов, которые в общем случае могут принадлежать разным сетям. Интерфейс, входящий в группу, получает наряду с обычным индивидуальным IP-адресом еще один групповой адрес. Если при отправке пакета в качестве адреса назначения указан адрес класса D, то такой пакет должен быть доставлен всем узлам, которые входят в группу.
- Если адрес начинается с последовательности 11110, то это значит, что данный адрес относится к **классу E**. Адреса этого класса зарезервированы для будущих применений.

Чтобы получить из IP-адреса номер сети и номер узла, требуется не только разделить адрес на две соответствующие части, но и дополнить каждую из них нулями до полных 4 байт. Возьмем, например, адрес класса B 129.64.134.5. Первые два байта идентифицируют сеть, а последующие два — узел. Таким образом, номером сети является адрес 129.64.0.0, а номером узла — адрес 0.0.134.5.

Особые IP-адреса

В TCP/IP существуют ограничения при назначении IP-адресов, а именно номера сетей и номера узлов *не могут состоять из одних двоичных нулей или единиц*. Отсюда следует, что максимальное количество узлов, приведенное в табл. 15.1 для сетей каждого класса, должно быть уменьшено на 2. Например, в адресах класса C под номер узла отводится 8 бит, которые позволяют задать 256 номеров: от 0 до 255. Однако в действительности максимальное число узлов в сети класса C не может превышать 254, так как адреса 0 и 255 запрещены для адресации сетевых интерфейсов. Из этих же соображений следует, что конечный узел

не может иметь адрес типа 98.255.255.255, поскольку номер узла в этом адресе класса A состоит из одних двоичных единиц.

Итак, некоторые IP-адреса интерпретируются особым образом:

- Если IP-адрес состоит только из двоичных нулей, то он называется **неопределенным адресом** и обозначает адрес того узла, который сгенерировал этот пакет. Адрес такого вида в особых случаях помещается в заголовок IP-пакета в поле адреса отправителя.
- Если в поле номера сети стоят только нули, то по умолчанию считается, что узел назначения принадлежит той же самой сети, что и узел, который отправил пакет. Такой адрес также может быть использован только в качестве адреса отправителя.
- Если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такой адрес называется **ограниченным широковещательным** (limited broadcast). Ограниченность в данном случае означает, что пакет не выйдет за границы данной сети не при каких условиях.
- Если в поле адреса назначения в разрядах, соответствующих номеру узла, стоят только единицы, то пакет, имеющий такой адрес, рассылается *всем* узлам сети, номер которой указан в адресе назначения. Например, пакет с адресом 192.190.21.255 будет направлен всем узлам сети 192.190.21.0. Такой тип адреса называется **широковещательным** (broadcast).

ВНИМАНИЕ

В протоколе IP нет понятия широковещания в том смысле, в котором оно используется в протоколах канального уровня локальных сетей, когда данные должны быть доставлены абсолютно всем узлам сети. Как ограниченный, так и обычный варианты широковещательной рассылки имеют пределы распространения в составной сети: они ограничены либо сетью, которой принадлежит источник пакета, либо сетью, номер которой указан в адресе назначения. Поэтому деление сети с помощью маршрутизаторов на части локализует широковещательный шторм пределами одной из подсетей просто потому, что нет способа адресовать пакет одновременно всем узлам всех сетей составной сети.

Особый смысл имеет IP-адрес, первый октет которого равен 127. Этот адрес является *внутренним адресом стека протоколов* компьютера (или маршрутизатора). Он используется для тестирования программ, а также для организации работы клиентской и серверной частей приложения, установленных на одном компьютере. Обе программные части данного приложения спроектированы в расчете на то, что они будут обмениваться сообщениями по сети. Но какой же IP-адрес они должны использовать для этого? Адрес сетевого интерфейса компьютера, на котором они установлены? Но это приводит к избыточным передачам пакетов в сеть. Экономичным решением является применение внутреннего адреса 127.0.0.0. В IP-сети запрещается присваивать сетевым интерфейсам IP-адреса, начинающиеся со значения 127. Когда программа посылает данные по IP-адресу 127.x.x.x, то данные не передаются в сеть, а возвращаются модулям верхнего уровня того же компьютера как только что принятые. Маршрут перемещения данных образует «петлю», поэтому этот адрес называется **адресом обратной петли** (loopback).

Уже упоминавшиеся групповые адреса, относящиеся к классу D, предназначены для экономичного распространения в Интернете или большой корпоративной сети аудио- или видеопрограмм, адресованных сразу большой аудитории слушателей или зрителей. Если групповой адрес помещен в поле адреса назначения IP-пакета, то данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу с номером, указанным

в поле адреса. Один и тот же узел может входить в несколько групп. В общем случае члены группы могут распределяться по различным сетям, находящимся друг от друга на произвольно большом расстоянии. Групповой адрес не делится на номера сети и узла и обрабатывается маршрутизатором особым образом. Основное назначение групповых адресов — распространение информации по схеме «один ко многим». От того, найдут групповые адреса широкое применение (сейчас их используют в основном небольшие экспериментальные «островки» в Интернете), зависит, сможет ли Интернет создать серьезную конкуренцию радио и телевидению.

Использование масок при IP-адресации

Снабжая каждый IP-адрес маской, можно отказаться от понятий классов адресов и сделать более гибкой систему адресации.

Пусть, например, для IP-адреса 129.64.134.5 указана маска 255.255.128.0, то есть в двоичном виде IP-адрес 129.64.134.5 — это:

10000001.01000000.10000110.00000101,

а маска 255.255.128.0 в двоичном виде выглядит так:

11111111.11111111.10000000.00000000.

Если игнорировать маску и интерпретировать адрес 129.64.134.5 на основе классов, то номером сети является 129.64.0.0, а номером узла — 0.0.134.5 (поскольку адрес относится к классу B).

Если же использовать маску, то 17 последовательных двоичных единиц в маске 255.255.128.0, «наложенные» на IP-адрес 129.64.134.5, делят его на две части, номер сети:

10000001.01000000.1

и номер узла:

0000110.00000101.

В десятичной форме записи номера сети и узла, дополненные нулями до 32 бит, выглядят соответственно как 129.64.128.0 и 0.0.6.5.

Наложение маски можно интерпретировать как выполнение логической операции И (AND). Так, в предыдущем примере номер сети из адреса 129.64.134.5 является результатом выполнения логической операции AND с маской 255.255.128.0:

10000001 01000000 10000110 00000101
AND
11111111.11111111.10000000.00000000

Для стандартных классов сетей маски имеют следующие значения:

- класс A — 11111111. 00000000. 00000000. 00000000 (255.0.0.0);
- класс B — 11111111. 11111111. 00000000. 00000000 (255.255.0.0);
- класс C — 11111111. 11111111. 11111111. 00000000 (255.255.255.0).

ПРИМЕЧАНИЕ

Для записи масок используются и другие форматы. Например, удобно интерпретировать значение маски, записанной в шестнадцатеричном коде: FFFF0000 — маска для адресов класса B. Еще чаще встречается обозначение 185.23.44.206/16 — данная запись говорит о том, что маска для этого адреса содержит 16 единиц или что в указанном IP-адресе под номер сети отведено 16 двоичных разрядов.

Механизм масок широко распространен в маршрутизации IP, причем маски могут использоваться для самых разных целей. С их помощью администратор может разбивать одну, выделенную ему поставщиком услуг сеть определенного класса на несколько других, не требуя от него дополнительных номеров сетей — эта операция называется *разделением на подсети* (subnetting). На основе этого же механизма поставщики услуг могут объединять адресные пространства нескольких сетей путем введения так называемых «префиксов» с целью уменьшения объема таблиц маршрутизации и повышения за счет этого производительности маршрутизаторов — такая операция называется *объединением подсетей* (supernetting). Подробнее об этом мы поговорим при изучении технологии бесклассовой междоменной маршрутизации.

Порядок назначения IP-адресов

По определению схема IP-адресации должна обеспечивать уникальность нумерации сетей, а также уникальность нумерации узлов в пределах каждой из сетей. Следовательно, процедуры назначения номеров как сетям, так и узлам сетей должны быть *централизованными*. Рекомендуемый порядок назначения IP-адресов дается в спецификации RFC 2050.

Назначение адресов автономной сети

Когда дело касается сети, являющейся частью Интернета, уникальность нумерации может быть обеспечена только усилиями специально созданных для этого центральных органов. В небольшой же автономной IP-сети условие уникальности номеров сетей и узлов может быть выполнено силами сетевого администратора.

В этом случае в распоряжении администратора имеется все адресное пространство, так как совпадение IP-адресов в не связанных между собой сетях не вызовет никаких отрицательных последствий. Администратор может выбирать адреса произвольным образом, соблюдая лишь синтаксические правила и учитывая ограничения на особые адреса. (Таким образом, номер узла в технологии TCP/IP назначается *независимо* от его локального адреса.)

Однако при таком подходе исключена возможность в будущем подсоединить данную сеть к Интернету. Действительно, произвольно выбранные адреса данной сети могут совпасть с централизовано назначенными адресами Интернета. Для того чтобы избежать коллизий, связанных с такого рода совпадениями, в стандартах Интернета определено несколько диапазонов так называемых **частных адресов**, рекомендуемых для автономного использования:

- в классе A — сеть 10.0.0.0;
- в классе B — диапазон из 16 номеров сетей (172.16.0.0–172.31.0.0);
- в классе C — диапазон из 255 сетей (192.168.0.0–192.168.255.0).

Эти адреса, исключенные из множества централизованно распределяемых, составляют огромное адресное пространство, достаточное для нумерации узлов автономных сетей практически любых размеров. Заметим также, что частные адреса, как и при произвольном выборе адресов, в разных автономных сетях могут совпадать. В то же время использование частных адресов для адресации автономных сетей делает возможным корректное подключение их к Интернету. Применяемые при этом специальные технологии подключения[1] исключают коллизии адресов.

[1] Например, такой технологией является NAT, которая рассматривается в главе 18.

Централизованное распределение адресов

В больших сетях, подобных Интернету, уникальность сетевых адресов гарантируется централизованной, иерархически организованной системой их распределения. Номер сети может быть назначен только по рекомендации специального подразделения Интернета. Главным органом регистрации глобальных адресов в Интернете с 1998 года является неправительственная некоммерческая организация **ICANN** (Internet Corporation for Assigned Names and Numbers). Эта организация координирует работу региональных отделов, деятельность которых охватывает большие географические площади: ARIN — Америка, RIPE (Европа), APNIC (Азия и Тихоокеанский регион). Региональные отделы выделяют блоки адресов сетей крупным поставщикам услуг, а те, в свою очередь, распределяют их между своими клиентами, среди которых могут быть и более мелкие поставщики.

Проблемой централизованного распределения адресов является их дефицит. Уже сравнительно давно очень трудно получить адрес класса B и практически невозможно стать обладателем адреса класса A. При этом надо отметить, что дефицит обусловлен не только ростом сетей, но и тем, что имеющееся адресное пространство используется нерационально. Очень часто владельцы сетей класса C расходуют лишь небольшую часть из имеющихся у них 254 адресов. Рассмотрим пример, когда две сети необходимо соединить глобальной связью. В таких случаях в качестве линии связи используют два маршрутизатора, соединенных по двухточечной схеме (рис. 15.4). Для вырожденной сети, образованной линией связи, связывающей порты двух смежных маршрутизаторов, приходится выделять отдельный номер сети, хотя в этой сети всего два узла.

Рис. 15.4. Нерациональное использование пространства IP-адресов

Для смягчения проблемы дефицита адресов разработчики стека TCP/IP предлагают разные подходы. Принципиальным решением является переход на новую версию протокола IP — протокол IPv6, в котором резко расширяется адресное пространство. Однако и текущая версия протокола IP (IPv4) поддерживает технологии, направленные на более экономное расходование IP-адресов, такие, например, как NAT и CIDR.

Адресация и технология CIDR

Технология **бесклассовой междоменной маршрутизации** (Classless Inter-Domain Routing, CIDR), которая описана в документах RFC 1517, RFC 1518, RFC 1519, RFC 1520 и о которой впервые было официально объявлено в 1993 году, позволяет центрам распределения адресов избежать выдачи абонентам излишних адресов.

Деление IP-адреса на номера сети и узла в технологии CIDR происходит на основе маски переменной длины, назначаемой поставщиком услуг. Непременным условием применимости CIDR является наличие у организации, распоряжающейся адресами, непрерывных диапазонов адресов. Такие адреса имеют одинаковый **префикс**, то есть одинаковую цифровую последовательность в нескольких старших разрядах. Пусть в распоряжении некоторого поставщика услуг имеется непрерывное пространство IP-адресов в количестве 2^n (рис. 15.5). Отсюда следует, что префикс имеет длину $(32 - n)$ разрядов. Оставшиеся n разрядов играют роль счетчика последовательных номеров.

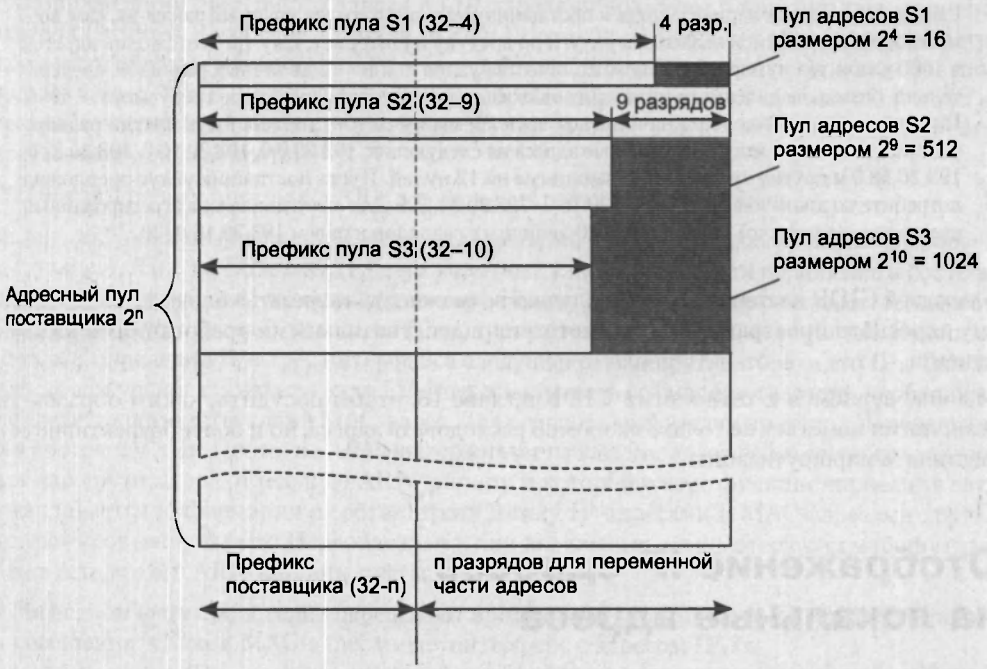


Рис. 15.5. Распределение адресов на основе технологии CIDR

Когда потребитель обращается к поставщику услуг с просьбой о выделении ему некоторого числа адресов, то в имеющемся пуле адресов «вырезается» непрерывная область *S1*, *S2* или *S3*, в зависимости от требуемого количества адресов. При этом должны быть выполнены следующие условия:

- количество адресов в выделяемой области должно быть равно степени двойки;
- начальная граница выделяемого пула адресов должна быть кратна требуемому количеству узлов.

Очевидно, что префикс каждой из показанных на рисунке областей имеет собственную длину — чем меньше количество адресов в данной области, тем длиннее ее префикс.

ПРИМЕР

Пусть поставщик услуг Интернета располагает пулом адресов в диапазоне 193.20.0.0–193.23.255.255 (1100 0001.0001 0100.0000 0000.0000 0000–1100 0001.0001 0111.1111 1111.1111 1111), то есть количество адресов равно 2^{18}. Соот-

ветственно префикс поставщика услуг имеет длину 14 разрядов — 1100 0001.0001 01, или в другом виде — 193.20/14.

Если абоненту этого поставщика услуг требуется совсем немного адресов, например 13, то поставщик мог бы предложить ему различные варианты: сеть 193.20.30.0/28, сеть 193.20.30.16/28 или сеть 193.21.204.48/28. Во всех случаях в распоряжении абонента для нумерации узлов имеются 4 младших бита. Таким образом, наименьшее число, удовлетворяющее потребностям абонента (13), которое можно представить степенью двойки (2^4), является 16. Префикс для каждого из выделяемых пулов во всех этих случаях играет роль номера сети, он имеет длину 32 – 4 = 28 разрядов.

Рассмотрим другой вариант, когда к поставщику услуг обратился крупный заказчик, сам, возможно, собирающийся оказывать услуги по доступу в Интернет. Ему требуется блок адресов в 4000 узлов. На нумерацию такого количества узлов пойдет 12 двоичных разрядов, следовательно, размер выделенного пула адресов оказывается несколько больше требуемого — 4096. Граница, с которой должен начинаться выделяемый участок, должна быть кратна размеру участка, то есть это могут быть любые адреса из следующих: 193.20.0.0, 193.20.16.0, 193.20.32.0, 193.20.48.0 и другие числа оканчивающиеся на 12 нулей. Пусть поставщик услуг предложил потребителю диапазон адресов 193.20.16.0–193.20.31.255. Для этого диапазона агрегированный номер сети (префикс) имеет длину 20 двоичных разрядов и равен 193.20.16.0/20.

Благодаря CIDR поставщик услуг получает возможность «нарезать» блоки из выделенного ему адресного пространства в соответствии с действительными требованиями каждого клиента.

Мы еще вернемся к технологии CIDR в главе 16, чтобы обсудить, каким образом эта технология помогает не только экономно расходовать адреса, но и более эффективно осуществлять маршрутизацию.

Отображение IP-адресов на локальные адреса

Одной из главных задач, которая ставилась при создании протокола IP, являлось обеспечение совместной согласованной работы в сети, состоящей из подсетей, в общем случае использующих разные сетевые технологии. Взаимодействие технологии TCP/IP с локальными технологиями подсетей происходит многократно при перемещении IP-пакета по составной сети. На каждом маршрутизаторе протокол IP определяет, какому следующему маршрутизатору в этой сети надо направить пакет. В результате решения этой задачи протоколу IP становится известен *IP-адрес* интерфейса следующего маршрутизатора (или конечного узла, если эта сеть является сетью назначения). Чтобы локальная технология сети смогла доставить пакет на следующий маршрутизатор, необходимо:

- упаковать пакет в кадр соответствующего для данной сети формата (например, Ethernet);
- снабдить данный кадр *локальным адресом* следующего маршрутизатора.

Решением этих задач, как уже отмечалось[1], занимается уровень сетевых интерфейсов стека TCP/IP.

[1] См. раздел «Стек протоколов TCP/IP» в главе 5.

Протокол разрешения адресов

Как уже было сказано, никакой функциональной зависимости между локальным адресом и его IP-адресом не существует, следовательно, единственный способ установления соответствия — ведение таблиц. В результате конфигурирования сети каждый интерфейс «знает» свои IP-адрес и локальный адрес, что можно рассматривать как таблицу, состоящую из одной строки. Проблема состоит в том, как организовать обмен имеющейся информацией между узлами сети.

Для определения локального адреса по IP-адресу используется **протокол разрешения адресов** (Address Resolution Protocol, ARP). Протокол разрешения адресов реализуется различным образом в зависимости от того, работает ли в данной сети протокол локальной сети (Ethernet, Token Ring, FDDI) с возможностью широковещания или же какой-либо из протоколов глобальной сети (Frame Relay, ATM), которые, как правило, не поддерживают широковещательный доступ.

Рассмотрим работу протокола ARP в локальных сетях с *широковещанием*.

На рис. 15.6 показан фрагмент IP-сети, включающий две сети — Ethernet1 (из трех конечных узлов *A*, *B* и *C*) и Ethernet2 (из двух конечных узлов *D* и *E*). Сети подключены соответственно к интерфейсам 1 и 2 маршрутизатора. Каждый сетевой интерфейс имеет IP-адрес и MAC-адрес. Пусть в какой-то момент IP-модуль узла *C* направляет пакет узлу *D*. Протокол IP узла *C* определил IP-адрес интерфейса следующего маршрутизатора — это IP_1. Теперь, прежде чем упаковать пакет в кадр Ethernet и направить его маршрутизатору, необходимо определить соответствующий MAC-адрес. Для решения этой задачи протокол IP обращается к протоколу ARP. Протокол ARP поддерживает на каждом интерфейсе сетевого адаптера или маршрутизатора отдельную ARP-таблицу, в которой в ходе функционирования сети накапливается информация о соответствии между IP-адресами и MAC-адресами других интерфейсов данной сети. Первоначально, при включении компьютера или маршрутизатора в сеть все его ARP-таблицы пусты.

1. На первом шаге происходит передача от протокола IP протоколу ARP примерно такого сообщения: «Какой MAC-адрес имеет интерфейс с адресом IP_1?»
2. Работа протокола ARP начинается с просмотра собственной ARP-таблицы. Предположим, что среди содержащихся в ней записей отсутствует запрашиваемый IP-адрес.
3. В этом случае исходящий IP-пакет, для которого оказалось невозможным определить локальный адрес из ARP-таблицы, запоминается в буфере, а протокол ARP формирует **ARP-запрос**, вкладывает его в кадр протокола Ethernet и широковещательно рассылает.
4. Все интерфейсы сети Ethernet1 получают ARP-запрос и направляют его «своему» протоколу ARP. ARP сравнивает указанный в запросе адрес IP_1 с IP-адресом интерфейса, на который поступил этот запрос. Протокол ARP, который констатировал совпадение (в данном случае это ARP маршрутизатора 1), формирует **ARP-ответ**.

В ARP-ответе маршрутизатор указывает локальный адрес MAC_1 своего интерфейса и отправляет его запрашивающему узлу (в данном примере узлу *C*), используя его локальный адрес. Широковещательный ответ в этом случае не требуется, так как формат ARP-запроса предусматривает поля локального и сетевого адресов отправителя. Заметим, что зона распространения ARP-запросов ограничивается сетью Ethernet1, так как на пути широковещательных кадров барьером стоит маршрутизатор.

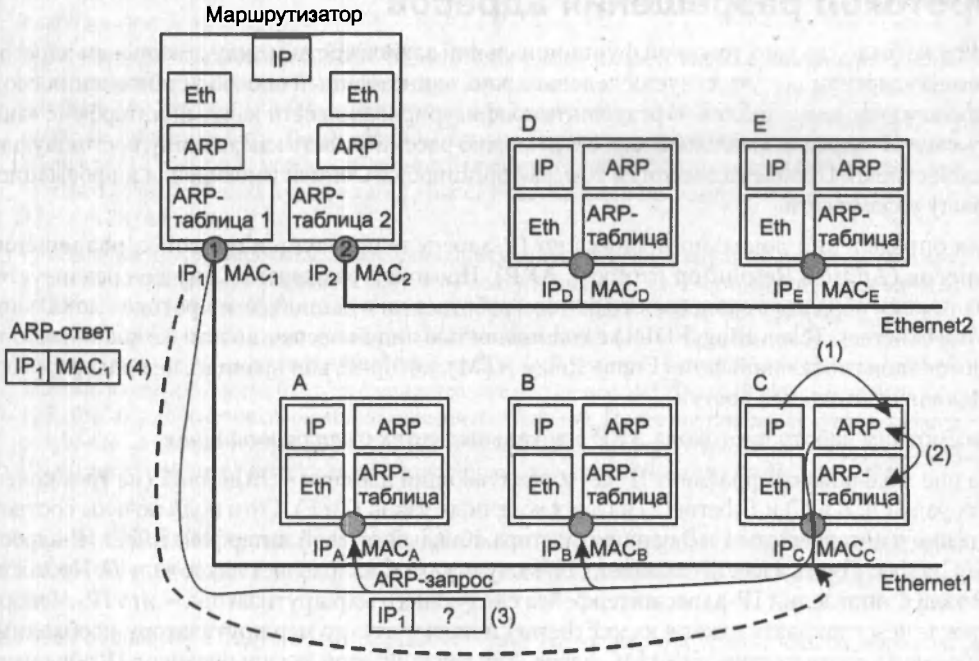


Рис. 15.6. Схема работы протокола ARP

На рис. 15.7 показан кадр Ethernet с вложенным в него ARP-сообщением. ARP-запросы и ARP-ответы имеют один и тот же формат. В табл. 15.2 в качестве примера приведены значения полей реального ARP-запроса, переданного по сети Ethernet[1].

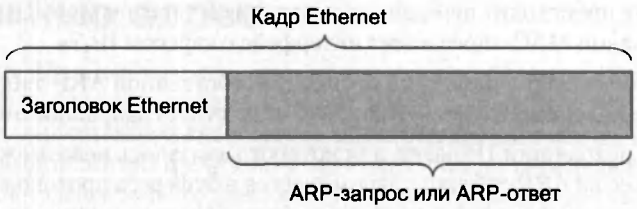


Рис. 15.7. Инкапсуляция ARP-сообщений в кадр Ethernet

В поле типа сети для сетей Ethernet указывается значение 1. Поле типа протокола позволяет использовать протокол ARP не только с протоколом IP, но и с другими сетевыми протоколами. Для IP значение этого поля равно 0x0800. Длина локального адреса для протокола Ethernet равна 6 байт, а длина IP-адреса — 4 байта. В поле операции для ARP-запросов указывается значение 1, для ARP-ответов — значение 2.

Из этого запроса видно, что в сети Ethernet узел с IP-адресом 194.85.135.75 пытается определить, какой MAC-адрес имеет другой узел той же сети, сетевой адрес которого 194.85.135.65. *Поле искомого локального адреса заполнено нулями.*

[1] Символы 0x означают, что за ними следует число, записанное в шестнадцатеричном формате.

Таблица 15.2. Пример ARP-запроса

Поле	Значение
Тип сети	1 (0x1)
Тип протокола	2048 (0x800)
Длина локального адреса	6 (0x6)
Длина сетевого адреса	4 (0x4)
Операция	1 (0x1)
Локальный адрес отправителя	008048EB7E60
Сетевой адрес отправителя	194.85.135.75
Локальный (искомый) адрес получателя	000000000000
Сетевой адрес получателя	194.85.135.65

Ответ присылает узел, опознавший свой IP-адрес. Если в сети нет машины с искомым IP-адресом, то ARP-ответа не будет. Протокол IP уничтожает IP-пакеты, направляемые по этому адресу. В табл. 15.3 показаны значения полей ARP-ответа, который мог бы поступить на приведенный в табл. 15.2 ARP-запрос.

Таблица 15.3. Пример ARP-ответа

Поле	Значение
Тип сети	1 (0x1)
Тип протокола	2048 (0x800)
Длина локального адреса	6 (0x6)
Длина сетевого адреса	4 (0x4)
Операция	2 (0x1)
Локальный адрес отправителя	00E0F77F1920
Сетевой адрес отправителя	194.85.135.65
Локальный (искомый) адрес получателя	008048EB7E60
Сетевой адрес получателя	194.85.135.75

В результате обмена ARP-сообщениями модуль IP, пославший запрос с интерфейса, имеющего адрес 194.85.135.75, определил, что IP-адресу 194.85.135.65 соответствует MAC-адрес 00E0F77F1920. Этот адрес затем помещается в заголовок кадра Ethernet, ожидавшего отправления IP-пакета.

Чтобы уменьшить число ARP-обращений в сети, найденное соответствие между IP-адресом и MAC-адресом сохраняется в ARP-таблице соответствующего интерфейса, в данном случае — это запись:

194.85.135.65 — 00E0F77F1920

Данная запись в ARP-таблице появляется автоматически, спустя несколько миллисекунд после того, как модуль ARP проанализирует ARP-ответ. Теперь, если вдруг вновь возникнет необходимость послать пакет по адресу 194.85.135.65, то протокол IP прежде, чем посылать широковещательный запрос, проверит, нет ли уже такого адреса в ARP-таблице.

ARP-таблица пополняется *не только за счет поступающих на данный интерфейс ARP-ответов*, но и в результате извлечения полезной информации из широковещательных ARP-запросов. Действительно, в каждом запросе, как это видно из табл. 15.2 и 15.3, содержатся IP-адрес и MAC-адрес отправителя. Все интерфейсы, получившие этот запрос, могут поместить информацию о соответствии локального и сетевого адресов отправителя в собственную ARP-таблицу. В частности, все узлы, получившие ARP-запрос (см. табл. 15.2), могут пополнить свою ARP-таблицу записью:

194.85.135.75 — 008048EB7E60

Таким образом, вид ARP-таблицы, в которую в ходе работы сети были добавлены две упомянутые нами записи, иллюстрирует табл. 15.4.

Таблица 15.4. Пример ARP-таблицы

IP-адрес	MAC-адрес	Тип записи
194.85.135.65	00E0F77F1920	Динамический
194.85.135.75	008048EB7E60	Динамический
194.85.60.21	008048EB7567	Статический

В ARP-таблицах существует два типа записей: динамические и статические. **Статические записи** создаются вручную с помощью утилиты arp и не имеют срока устаревания, точнее, они существуют до тех пор, пока компьютер или маршрутизатор остается включенным. **Динамические записи** должны периодически обновляться. Если запись не обновлялась в течение определенного времени (порядка нескольких минут), то она исключается из таблицы. Таким образом, в ARP-таблице содержатся записи не обо всех узлах сети, а только о тех, которые активно участвуют в сетевых операциях. Поскольку такой способ хранения информации называют кэшированием, ARP-таблицы иногда называют **ARP-кэшем**.

ПРИМЕЧАНИЕ

Некоторые реализации протоколов IP и ARP не ставят IP-пакеты в очередь на время ожидания ARP-ответов. Вместо этого IP-пакет просто уничтожается, а его восстановление возлагается на модуль TCP или прикладной процесс, работающий через протокол UDP. Такое восстановление выполняется за счет тайм-аутов и повторных передач. Повторная передача сообщения проходит успешно, так как первая попытка уже вызвала заполнение ARP-таблицы.

Совсем другой способ разрешения адресов используется *в глобальных сетях*, в которых не поддерживается широковещательная рассылка. Здесь администратору сети чаще всего приходится вручную формировать и помещать на какой-либо сервер ARP-таблицы, в которых он задает, например, соответствие IP-адресов адресам X.25, имеющих для протокола IP смысл локальных адресов. В то же время сегодня наметилась тенденция автоматизации работы протокола ARP и в глобальных сетях. Для этой цели среди всех маршрутизаторов, подключенных к какой-либо глобальной сети, выделяется специальный маршрутизатор, который ведет ARP-таблицу для всех остальных узлов и маршрутизаторов этой сети.

При таком централизованном подходе вручную нужно задать для всех узлов и маршрутизаторов только IP-адрес и локальный адрес выделенного для этих целей маршрутизатора. При включении каждый узел и маршрутизатор регистрирует свои адреса в выделенном маршрутизаторе. Всякий раз, когда возникает необходимость определения по IP-адресу локального адреса, модуль ARP обращается к выделенному маршрутизатору с запросом

и автоматически получает ответ без участия администратора. Работающий таким образом маршрутизатор называют **ARP-сервером**.

В некоторых случаях возникает обратная задача — нахождение IP-адреса по известному локальному адресу. Тогда в действие вступает *реверсивный протокол разрешения адресов* (Reverse Address Resolution Protocol, RARP). Этот протокол используется, например, при старте бездисковых станций, не знающих в начальный момент времени своего IP-адреса, но знающих MAC-адрес своего сетевого адаптера.

Протокол Proxy-ARP

Протокол Proxy-ARP — это одна из разновидностей протокола ARP, позволяющая отображать IP-адреса на аппаратные адреса в сетях, поддерживающих широковещание, даже в тех случаях, когда искомый узел находится за пределами данного домена коллизий.

На рис. 15.8 показана сеть, один из конечных узлов которой (компьютер *D*) работает в режиме удаленного узла. Подробнее об этом режиме рассказывается в главе 22, а сейчас достаточно знать, что конечный узел в таком режиме обладает всеми возможностями компьютеров, работающих в основной части сети Ethernet, в частности он имеет IP-адрес (IP_D), относящийся к той же сети. Для всех конечных узлов сети Ethernet особенности подключения удаленного узла (наличие модемов, коммутируемая связь, протокол PPP) абсолютно прозрачны — они взаимодействуют с ним обычным образом. Чтобы такой режим взаимодействия стал возможным, среди прочего, необходим протокол Proxy-ARP. Поскольку удаленный узел подключен к сети по протоколу PPP, то он, очевидно, *не имеет MAC-адреса.*

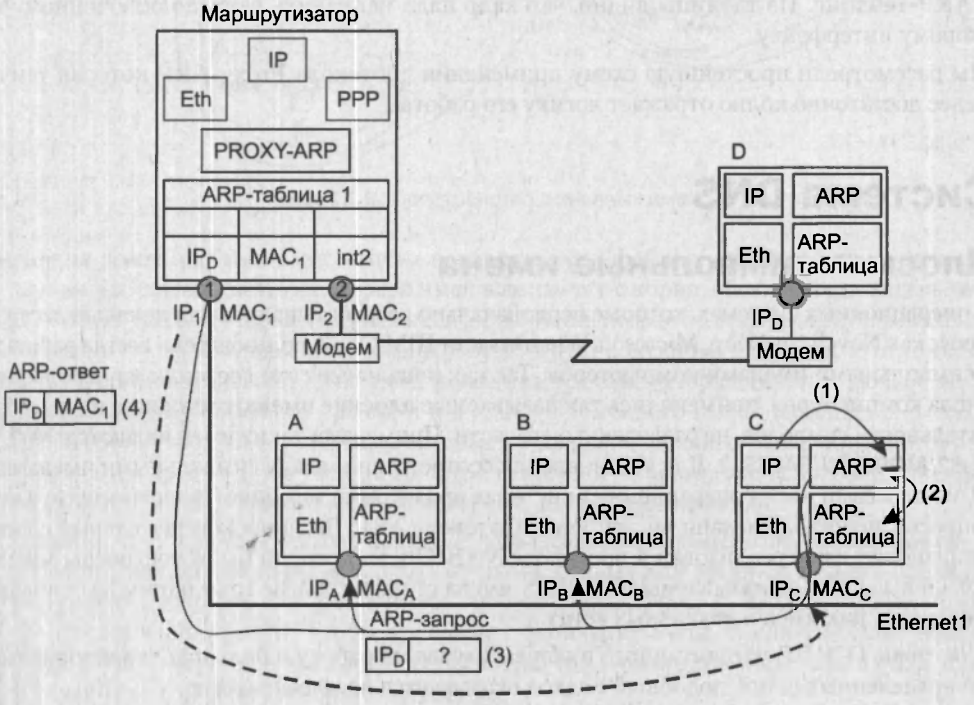


Рис. 15.8. Схема работы протокола Proxy-ARP

Пусть приложение, работающее, например, на компьютере *C*, решает послать пакет компьютеру *D*. Ему известен IP-адрес узла назначения (IP_D), однако, как мы уже не раз отмечали, для передачи пакета по сети Ethernet его необходимо упаковать в кадр Ethernet и снабдить MAC-адресом. Для определения MAC-адреса IP-протокол узла *C* обращается к протоколу ARP, который посылает широковещательное сообщение с ARP-запросом. Если бы в этой сети на маршрутизаторе не был установлен протокол Proxy-ARP, на этот запрос не откликнулся бы ни один узел.

Однако протокол Proxy-ARP установлен на маршрутизаторе и работает следующим образом. При подключении к сети удаленного узла *D* в таблицу ARP-маршрутизатора заносится запись

$$IP_D - MAC_1 - int2,$$

которая означает, что:

- при поступлении ARP-запроса на маршрутизатор относительно адреса IP*D* в ARP-ответ будет помещен аппаратный адрес MAC_1, соответствующий аппаратному адресу интерфейса 1 маршрутизатора;
- узел, имеющий адрес IP_D, подключен к интерфейсу 2 маршрутизатора.

В ответ на посланный узлом *C* широковещательный ARP-запрос откликается маршрутизатор с установленным протоколом Proxy-ARP. Он посылает «ложный» ARP-ответ, в котором на место аппаратного адреса компьютера *D* помещает собственный адрес MAC_1. Узел *C*, не подозревая «подвоха», посылает кадр с IP-пакетом по адресу MAC_1. Получив кадр, маршрутизатор с установленным протоколом Proxy-ARP «понимает», что он направлен не ему (в пакете указан чужой IP-адрес) и, следовательно, надо искать адресата в ARP-таблице. Из таблицы видно, что кадр надо направить узлу, подключенному ко второму интерфейсу.

Мы рассмотрели простейшую схему применения протокола Proxy-ARP, которая тем не менее достаточно полно отражает логику его работы.

Система DNS

Плоские символьные имена

В операционных системах, которые первоначально разрабатывались для локальных сетей, таких как Novell NetWare, Microsoft Windows или IBM OS/2, пользователи всегда работали с символьными именами компьютеров. Так как локальные сети состояли из небольшого числа компьютеров, применялись так называемые **плоские имена**, состоящие из последовательности символов, не разделенных на части. Примерами таких имен являются: NW1_1, mail2, MOSCOW_SALES_2. Для установления соответствия между символьными именами и MAC-адресами в этих операционных системах применялся механизм широковещательных запросов, подобный механизму запросов протокола ARP. Так, широковещательный способ разрешения имен реализован в протоколе NetBIOS, на котором были построены многие локальные ОС. Так называемые NetBIOS-имена стали на долгие годы одним из основных типов плоских имен в локальных сетях.

Для стека TCP/IP, рассчитанного в общем случае на работу в больших территориально распределенных сетях, подобный подход оказывается неэффективным.

Иерархические символьные имена

В стеке TCP/IP применяется доменная система имен, которая имеет иерархическую древовидную структуру, допускающую наличие в имени произвольного количества составных частей (рис. 15.9).

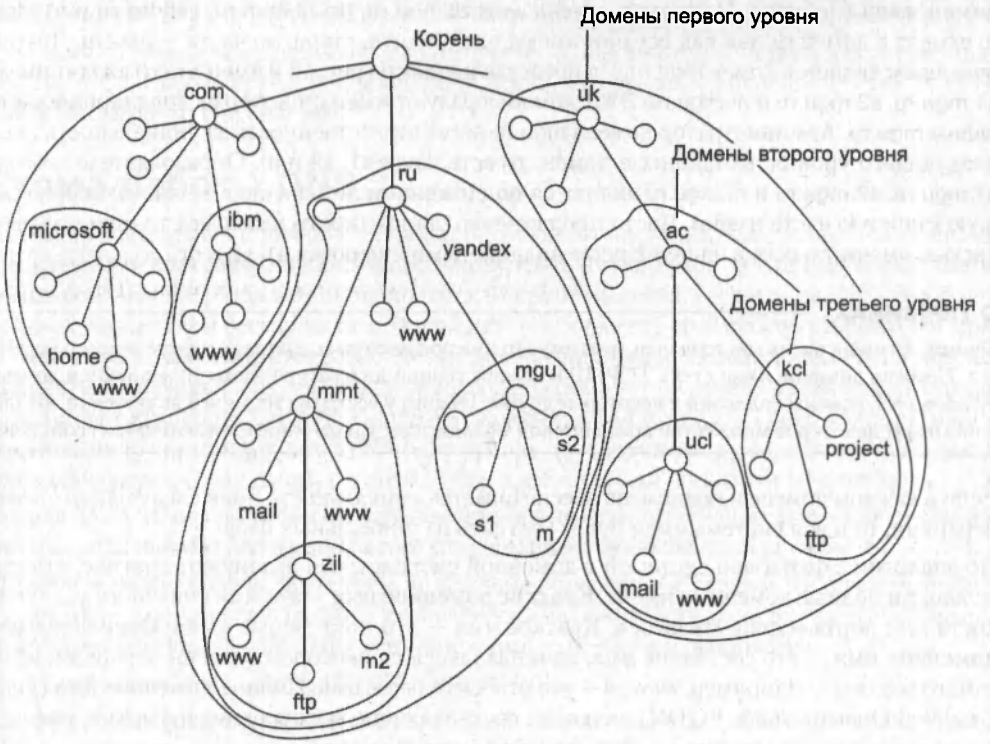


Рис. 15.9. Пространство доменных имен

Иерархия доменных имен аналогична иерархии имен файлов, принятой во многих популярных файловых системах. Дерево имен начинается с корня, обозначаемого здесь точкой (.). Затем следует старшая символьная часть имени, вторая по старшинству символьная часть имени и т. д. Младшая часть имени соответствует конечному узлу сети. В отличие от имен файлов, при записи которых сначала указывается самая старшая составляющая, затем составляющая более низкого уровня и т. д., запись доменного имени начинается с самой младшей составляющей, а заканчивается самой старшей. Составные части доменного имени отделяются друг от друга точкой. Например, в имени home.microsoft.com составляющая home является именем одного из компьютеров в домене microsoft.com.

Разделение имени на части позволяет *разделить административную ответственность* за назначение уникальных имен между различными людьми или организациями в пределах своего уровня иерархии. Так, для примера, приведенного на рис. 15.9, один человек может нести ответственность за то, чтобы все имена с окончанием «ru» имели уникальную следующую вниз по иерархии часть. То есть все имена типа www.ru, mail.mmt.ru или m2.zil.mmt.ru отличаются второй по старшинству частью.

Разделение административной ответственности позволяет решить проблему образования уникальных имен без взаимных консультаций между организациями, отвечающими за имена одного уровня иерархии. Очевидно, что должна существовать одна организация, отвечающая за назначение имен верхнего уровня иерархии.

Совокупность имен, у которых несколько старших составных частей совпадают, образуют **домен имен** (domain). Например, имена www.zil.mmt.ru, ftp.zil.mmt.ru, yandex.ru и s1.mgu.ru входят в домен ru, так как все они имеют одну общую старшую часть — имя ru. Другим примером является домен mgu.ru. Из представленных на рис. 15.9 имен в него входят имена s1.mgu.ru, s2.mgu.ru и m.mgu.ru. Этот домен образуют имена, у которых две старшие части равны mgu.ru. Администратор домена mgu.ru несет ответственность за уникальность имен следующего уровня, входящих в домен, то есть имен s1, s2 и m. Образованные домены s1.mgu.ru, s2.mgu.ru и m.mgu.ru являются **поддоменами** домена mgu.ru, так как имеют общую старшую часть имени. Часто проддомены для краткости называют только младшей частью имени, то есть в нашем случае поддоменами являются s1, s2 и m.

О ТЕРМИНАХ

Термин «домен» очень многозначен, поэтому его нужно трактовать в рамках определенного контекста. Помимо доменов имен стека TCP/IP в компьютерной литературе часто упоминаются домены Windows NT, домены коллизий и некоторые другие. Общим у всех этих терминов является то, что они описывают некоторое множество компьютеров, обладающее каким-либо определенным свойством.

Если в каждом домене и поддомене обеспечивается уникальность имен следующего уровня иерархии, то и вся система имен будет состоять из уникальных имен.

По аналогии с файловой системой в доменной системе имен различают краткие, относительные и полные доменные имена. **Краткое доменное имя** — это имя конечного узла сети: хоста или порта маршрутизатора. Краткое имя — это лист дерева имен. **Относительное доменное имя** — это составное имя, начинающееся с некоторого уровня иерархии, но не самого верхнего. Например, www.zil — это относительное имя. **Полное доменное имя** (Fully Qualified Domain Name, FQDN) включает составляющие всех уровней иерархии, начиная от краткого имени и кончая корневой точкой: www.zil.mmt.ru.

ВНИМАНИЕ

Компьютеры, имена которых относятся к одному и тому же домену, могут иметь абсолютно независимые друг от друга IP-адреса, принадлежащие различным сетям и подсетям. Например, в домен mgu.ru могут входить хосты с адресами 132.13.34.15, 201.22.100.33 и 14.0.0.6.

Корневой домен управляется центральными органами Интернета, в частности уже упоминавшейся нами организацией ICANN. Домены верхнего уровня назначаются для каждой страны, а также для различных типов организаций. Имена этих доменов должны следовать международному стандарту ISO 3166. Для обозначения стран используются трехбуквенные и двухбуквенные аббревиатуры, например ru (Россия), uk (Великобритания), fi (Финляндия), us (Соединенные Штаты), а для различных типов организаций, например, следующие обозначения:

- com — коммерческие организации (например, microsoft.com);
- edu — образовательные организации (например, mit.edu);
- gov — правительственные организации (например, nsf.gov);

- org — некоммерческие организации (например, fidonet.org);
- net — сетевые организации (например, nsf.net).

Каждый домен администрирует отдельная организация, которая обычно разбивает свой домен на поддомены и передает функции администрирования этих поддоменов другим организациям. Чтобы получить доменное имя, необходимо зарегистрироваться в какой-либо организации, которой делегированы полномочия по распределению имен доменов.

Доменная система имен реализована в Интернете, но она может работать и как автономная система имен в любой крупной корпоративной сети, которая хотя и использует стек TCP/IP, никак не связана с Интернетом.

Схема работы DNS

Широковещательный способ установления соответствия между символьными именами и локальными адресами, подобный протоколу ARP, хорошо работает только в небольшой локальной сети, не разделенной на подсети. В крупных сетях, где возможность всеобщей широковещательной рассылки не поддерживается, нужен другой способ разрешения символьных имен. Хорошей альтернативой широковещательной рассылке является применение централизованной службы, поддерживающей соответствие между различными типами адресов всех компьютеров сети. Например, компания Microsoft для своей корпоративной операционной системы Windows NT разработала централизованную службу WINS, которая поддерживала базу данных NetBIOS-имен и соответствующих им IP-адресов.

В сетях TCP/IP соответствие между доменными именами и IP-адресами может устанавливаться средствами как локального хоста, так и централизованной службы.

На раннем этапе развития Интернета на каждом хосте вручную создавался текстовый файл с известным именем hosts.txt. Этот файл состоял из некоторого количества строк, каждая из которых содержала одну пару «доменное имя — IP-адрес», например:

rhino.acme.com — 102.54.94.97

По мере роста Интернета файлы hosts.txt также увеличивались в объеме, и создание *масштабируемого* решения для разрешения имен стало необходимостью.

Таким решением стала *централизованная служба* **DNS** (Domain Name System — система доменных имен), основанная на распределенной базе отображений «доменное имя — IP-адрес». Служба DNS использует в своей работе DNS-серверы и DNS-клиенты. DNS-серверы поддерживают распределенную базу отображений, а DNS-клиенты обращаются к серверам с запросами об отображении разрешении доменного имени на IP-адрес.

Служба DNS использует текстовые файлы почти такого же формата, как и файл hosts, и эти файлы администратор также подготавливает вручную. Однако служба DNS опирается на иерархию доменов, и каждый DNS-сервер хранит только часть имен сети, а не все имена, как это происходит при использовании файлов hosts. При росте количества узлов в сети проблема масштабирования решается созданием новых доменов и поддоменов имен и добавлением в службу DNS новых серверов.

Для каждого домена имен создается свой DNS-сервер. На серверах применяют два подхода к распределению имен. В первом случае сервер может хранить отображения «доменное имя — IP-адрес» для всего домена, включая все его поддомены. Однако такое решение оказывается плохо масштабируемым, так как при добавлении новых поддоменов нагрузка на этот сервер может превысить его возможности. Чаще используется другой подход, когда

сервер домена хранит только имена, которые заканчиваются на следующем ниже уровне иерархии по сравнению с именем домена. (Аналогично каталогу файловой системы, который содержит записи о файлах и подкаталогах, непосредственно в него «входящих».) Именно при такой организации службы DNS нагрузка по разрешению имен распределяется более-менее равномерно между всеми DNS-серверами сети. Например, в первом случае DNS-сервер домена mmt.ru будет хранить отображения для всех имен, заканчивающихся на mmt.ru (www1.zil.mmt.ru, ftp.zil.mmt.ru, mail.mmt.ru и т. д.). Во втором случае этот сервер хранит отображения только имен типа mail.mmt.ru, www.mmt.ru, а все остальные отображения должны храниться на DNS-сервере поддомена zil.

Каждый DNS-сервер помимо таблицы отображений имен содержит ссылки на DNS-серверы своих поддоменов. Эти ссылки связывают отдельные DNS-серверы в единую службу DNS. Ссылки представляют собой IP-адреса соответствующих серверов. Для обслуживания корневого домена выделено несколько дублирующих друг друга DNS-серверов, IP-адреса которых широко известны (их можно узнать, например, в InterNIC).

Процедура разрешения DNS-имени во многом аналогична процедуре поиска файловой системой адреса файла по его символьному имени. Действительно, в обоих случаях составное имя отражает иерархическую структуру организации соответствующих справочников — каталогов файлов или DNS-таблиц. Здесь домен и доменный DNS-сервер являются аналогом каталога файловой системы. Для доменных имен, так же как и для символьных имен файлов, характерна независимость именования от физического местоположения.

Процедура поиска адреса файла по символьному имени заключается в последовательном просмотре каталогов, начиная с корневого. При этом предварительно проверяются кэш и текущий каталог. Для определения IP-адреса по доменному имени также необходимо просмотреть все DNS-серверы, обслуживающие цепочку поддоменов, входящих в имя хоста, начиная с корневого домена.

Существенным отличием файловой системы от службы DNS является то, что первая расположена на одном компьютере, а вторая по своей природе является *распределенной*.

Существует две основные схемы разрешения DNS-имен. В первом варианте работу по поиску IP-адреса координирует DNS-клиент:

1. DNS-клиент обращается к корневому DNS-серверу с указанием полного доменного имени.
2. DNS-сервер отвечает клиенту, указывая адрес следующего DNS-сервера, обслуживающего домен верхнего уровня, заданный в следующей старшей части запрошенного имени.
3. DNS-клиент делает запрос следующего DNS-сервера, который отсылает его к DNS-серверу нужного поддомена и т. д., пока не будет найден DNS-сервер, в котором хранится соответствие запрошенного имени IP-адресу. Этот сервер дает окончательный ответ клиенту.

Такая процедура разрешения имени называется **нерекурсивной**, когда клиент сам итеративно выполняет последовательность запросов к разным серверам имен. Эта схема загружает клиента достаточно сложной работой, и она применяется редко.

Во втором варианте реализуется **рекурсивная** процедура:

1. DNS-клиент запрашивает локальный DNS-сервер, то есть тот сервер, обслуживающий поддомен, которому принадлежит имя клиента.

2. Далее возможны два варианта действий:
   - если локальный DNS-сервер знает ответ, то он сразу же возвращает его клиенту (это может произойти, когда запрошенное имя входит в тот же поддомен, что и имя клиента, или когда сервер уже узнавал данное соответствие для другого клиента и сохранил его в своем кэше);
   - если локальный сервер не знает ответ, то он выполняет итеративные запросы к корневому серверу и т. д. точно так же, как это делал клиент в предыдущем варианте, а получив ответ, передает его клиенту, который все это время просто ждет его от своего локального DNS-сервера.

В этой схеме клиент перепоручает работу своему серверу, именно поэтому схема называется рекурсивной, или косвенной. Практически все DNS-клиенты используют рекурсивную процедуру.

Для ускорения поиска IP-адресов DNS-серверы широко применяют *кэширование* проходящих через них ответов. Чтобы служба DNS могла оперативно отрабатывать изменения, происходящие в сети, ответы кэшируются на относительно короткое время — обычно от нескольких часов до нескольких дней.

Обратная зона

Служба DNS предназначена не только для нахождения IP-адреса по имени хоста, но и для решения *обратной задачи* — нахождению DNS-имени по известному IP-адресу.

Многие программы и утилиты, пользующиеся службой DNS, пытаются найти имя узла по его адресу в том случае, когда пользователем задан только адрес (или этот адрес программа узнала из пришедшего пакета). Обратная запись не всегда существует даже для тех адресов, для которых есть прямые записи. Ее могут просто забыть создать или же ее создание требует дополнительной оплаты. Обратная задача решается в Интернете путем организации так называемых обратных зон.

Обратная зона — это система таблиц, которая хранит соответствие между IP-адресами и DNS-имена хостов некоторой сети. Для организации распределенной службы и использования для поиска имен того же программного обеспечения, что и для поиска адресов, применяется оригинальный подход, связанный с представлением IP-адреса в виде DNS-имени.

Первый этап преобразования заключается в том, что составляющие IP-адреса интерпретируются как составляющие DNS-имени. Например, адрес 192.31.106.0 рассматривается как состоящий из старшей части, соответствующей домену 192, затем идет домен 31, в который входит домен 106.

Далее, учитывая, что при записи IP-адреса старшая часть является самой *левой* частью адреса, а при записи DNS-имени — самой *правой*, то составляющие в преобразованном адресе указываются в обратном порядке, то есть для данного примера — 106.31.192.

Для хранения соответствия всех адресов, начинающихся, например, с числа 192, заводится зона 192 со своими серверами имен. Для записей о серверах, поддерживающих старшие в иерархии обратные зоны, создана специальная зона in-addr.arpa, поэтому полная запись для использованного в примере адреса выглядит так:

```
106.31.192.in-addr.arpa.
```

Серверы для обратных зон используют файлы баз данных, не зависящие от файлов основных зон, в которых имеются записи о прямом соответствии тех же имен и адресов. Такая организация данных может приводить к несогласованности, так как одно и то же соответствие вводится в файлы дважды.

Протокол DHCP

Для нормальной работы сети каждому сетевому интерфейсу компьютера и маршрутизатора должен быть назначен IP-адрес.

Процедура присвоения адресов происходит в ходе **конфигурирования** компьютеров и маршрутизаторов. Назначение IP-адресов может происходить вручную в результате выполнения процедуры конфигурирования интерфейса, для компьютера сводящейся, например, к заполнению системы экранных форм. При этом администратор должен помнить, какие адреса из имеющегося множества он уже использовал для других интерфейсов, а какие еще свободны. При конфигурировании помимо IP-адресов сетевых интерфейсов (и соответствующих масок) устройству сообщается ряд других **конфигурационных параметров**. При конфигурировании администратор должен назначить клиенту не только IP-адрес, но и другие параметры стека TCP/IP, необходимые для его эффективной работы, например маску и IP-адрес маршрутизатора по умолчанию, IP-адрес DNS-сервера, доменное имя компьютера и т. п. Даже при не очень большом размере сети эта работа представляет для администратора утомительную процедуру.

Протокол динамического конфигурирования хостов (Dynamic Host Configuration Protocol, DHCP) автоматизирует процесс конфигурирования сетевых интерфейсов, обеспечивая отсутствие дублирования адресов за счет централизованного управления их распределением. Работа DHCP описана в RFC 2131 и 2132.

Режимы DHCP

Протокол DHCP работает в соответствии с моделью *клиент-сервер*. Во время старта системы компьютер, являющийся DHCP-клиентом, посылает в сеть широковещательный запрос на получение IP-адреса. DHCP-сервер откликается и посылает сообщение-ответ, содержащее IP-адрес и некоторые другие конфигурационные параметры.

При этом сервер DHCP может работать в разных режимах, включая:

- ручное назначение статических адресов;
- автоматическое назначение статических адресов;
- автоматическое распределение динамических адресов.

Во всех режимах работы администратор при конфигурировании DHCP-сервера сообщает ему один или несколько диапазонов IP-адресов, причем все эти адреса относятся к одной сети, то есть имеют одно и то же значение в поле номера сети.

В **ручном** режиме администратор, помимо пула доступных адресов, снабжает DHCP-сервер информацией о жестком соответствии IP-адресов физическим адресам или другим идентификаторам клиентских узлов. DHCP-сервер, пользуясь этой информацией, *всегда*

выдаст определенному DHCP-клиенту *один и тот же* назначенный ему администратором IP-адрес (а также набор других конфигурационных параметров[1]).

В режиме **автоматического** назначения статических адресов DHCP-сервер самостоятельно без вмешательства администратора произвольным образом выбирает клиенту IP-адрес из пула наличных IP-адресов. Адрес дается клиенту из пула в постоянное пользование, то есть между идентифицирующей информацией клиента и его IP-адресом по-прежнему, как и при ручном назначении, существует постоянное соответствие. Оно устанавливается в момент первого назначения DHCP-сервером IP-адреса клиенту. При всех последующих запросах сервер возвращает клиенту тот же самый IP-адрес.

При **динамическом** распределении адресов DHCP-сервер выдает адрес клиенту на ограниченное время, называемое **сроком аренды**. Когда компьютер, являющийся DHCP-клиентом, удаляется из подсети, назначенный ему IP-адрес автоматически освобождается. Когда компьютер подключается к другой подсети, то ему автоматически назначается новый адрес. Ни пользователь, ни сетевой администратор не вмешиваются в этот процесс.

Это дает возможность впоследствии повторно использовать этот IP-адрес для назначения другому компьютеру. Таким образом, помимо основного преимущества DHCP – автоматизации рутинной работы администратора по конфигурированию стека TCP/IP на каждом компьютере, режим динамического распределения адресов в принципе позволяет строить IP-сеть, количество узлов в которой превышает количество имеющихся в распоряжении администратора IP-адресов.

ПРИМЕР

Рассмотрим преимущества, которые дает динамическое распределение пула адресов на примере организации, в которой сотрудники значительную часть рабочего времени проводят вне офиса — дома или в командировках. Каждый из них имеет портативный компьютер, который во время пребывания в офисе подключается к корпоративной IP-сети. Возникает вопрос, сколько IP-адресов необходимо этой организации?

Первый ответ — столько, *скольким сотрудникам необходим доступ в сеть*. Если их 500 человек, то каждому из них должен быть назначен IP-адрес и выделено рабочее место. То есть администрация должна получить у поставщика услуг адреса двух сетей класса C и оборудовать соответствующим образом помещение. Однако вспомним, что сотрудники в этой организации редко появляются в офисе, значит, большая часть ресурсов при таком решении будет простаивать.

Второй ответ — столько, *сколько сотрудников обычно присутствует в офисе* (с некоторым запасом). Если обычно в офисе работает не более 50 сотрудников, то достаточно получить у поставщика услуг пул из 64 адресов и установить в рабочем помещении сеть с 64-я коннекторами для подключения компьютеров. Но возникает другая проблема — кто и как будет конфигурировать компьютеры, состав которых постоянно меняется?

Существует два пути. Во-первых, администратор (или сам мобильный пользователь) может конфигурировать компьютер вручную каждый раз, когда возникает необходимость подключения к офисной сети. Такой подход требует от администратора (или пользователей) большого объема рутинной работы, следовательно — это плохое решение. Гораздо привлекательнее выглядят возможности автоматического динамического назначения DHCP-адресов. Действительно, администратору достаточно один раз при настройке DHCP-сервера указать диапазон из 64 адресов, а каждый вновь прибывающий мобильный пользователь будет просто физически подключать в сеть свой компьютер, на котором запускается DHCP-клиент.

[1] Иногда мы будем для краткости опускать это уточнение.

Он запросит конфигурационные параметры и автоматически получит их от DHCP-сервера. Таким образом, для работы 500 мобильных сотрудников достаточно иметь в офисной сети 64 IP-адреса и 64 рабочих места.

Алгоритм динамического назначения адресов

Администратор управляет процессом конфигурирования сети, определяя два основных конфигурационных параметра DHCP-сервера: *пул адресов, доступных распределению*, и *срок аренды*. Срок аренды диктует, как долго компьютер может использовать назначенный IP-адрес, перед тем как снова запросить его от DHCP-сервера. Срок аренды зависит от режима работы пользователей сети. Если это небольшая сеть учебного заведения, куда со своими компьютерами приходят многочисленные студенты для выполнения лабораторных работ, то срок аренды может быть равен длительности лабораторной работы. Если же это корпоративная сеть, в которой сотрудники предприятия работают на регулярной основе, то срок аренды может быть достаточно длительным — несколько дней или даже недель.

DHCP-сервер должен находиться в одной подсети с клиентами, учитывая, что клиенты посылают ему широковещательные запросы (рис. 15.10). Для снижения риска выхода сети из строя из-за отказа DHCP-сервера в сети иногда ставят резервный DHCP-сервер (такой вариант соответствует сети 1).

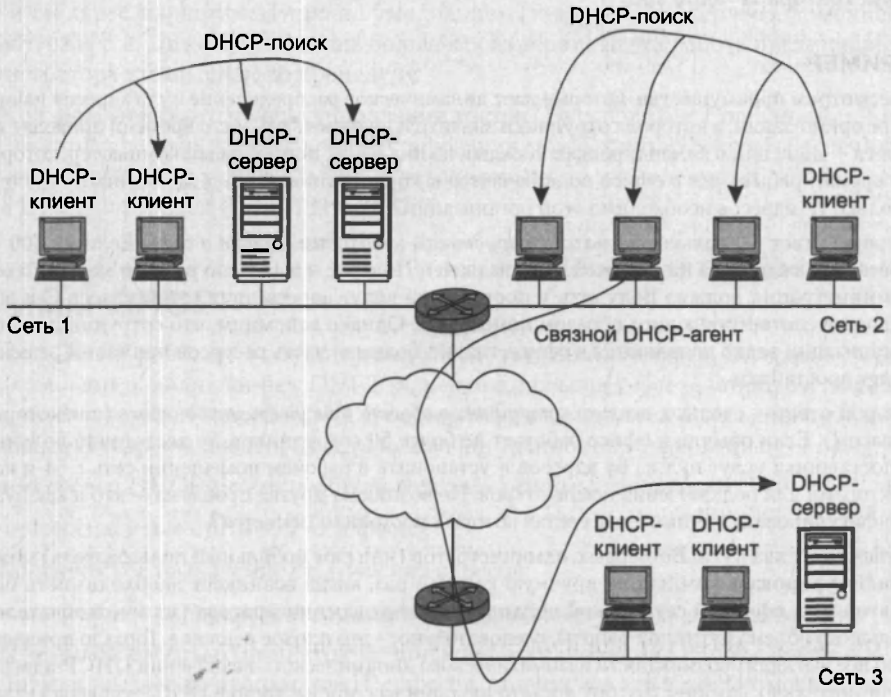


Рис. 15.10. Схемы взаимного расположения DHCP-серверов и DHCP-клиентов

Иногда наблюдается и обратная картина: в сети нет ни одного DHCP-сервера. В этом случае его подменяет связной **DHCP-агент** — программное обеспечение, играющее роль

посредника между DHCP-клиентами и DHCP-серверами (пример такого варианта — сеть 2). Связной агент переправляет запросы клиентов из сети 2 DHCP-серверу сети 3. Таким образом, один DHCP-сервер может обслуживать DHCP-клиентов нескольких разных сетей.

Вот как выглядит упрощенная схема обмена сообщениями между клиентскими и серверными частями DHCP.

1. Когда компьютер включают, установленный на нем DHCP-клиент посылает ограниченное широковещательное сообщение DHCP-поиска (IP-пакет с адресом назначения, состоящим из одних единиц, который должен быть доставлен всем узлам данной IP-сети).
2. Находящиеся в сети DHCP-серверы получают это сообщение. Если в сети DHCP-серверы отсутствуют, то сообщение DHCP-поиска получает связной DHCP-агент. Он пересылает это сообщение в другую, возможно, значительно отстоящую от него сеть DHCP-серверу, IP-адрес которого ему заранее известен.
3. Все DHCP-серверы, получившие сообщение DHCP-поиска, посылают DHCP-клиенту, обратившемуся с запросом, свои DHCP-предложения. Каждое предложение содержит IP-адрес и другую конфигурационную информацию. (DHCP-сервер, находящийся в другой сети, посылает ответ через агента.)
4. DHCP-клиент собирает конфигурационные DHCP-предложения от всех DHCP-серверов. Как правило, он выбирает первое из поступивших предложений и отправляет в сеть широковещательный DHCP-запрос. В этом запросе содержатся идентификационная информация о DHCP-сервере, предложение которого принято, а также значения принятых конфигурационных параметров.
5. Все DHCP-серверы получают DHCP-запрос, и только один выбранный DHCP-сервер посылает положительную DHCP-квитанцию (подтверждение IP-адреса и параметров аренды), а остальные серверы аннулируют свои предложения, в частности возвращают в свои пулы предложенные адреса.
6. DHCP-клиент получает положительную DHCP-квитанцию и переходит в рабочее состояние.

Время от времени компьютер пытается обновить параметры аренды у DHCP-сервера. Первую попытку он делает задолго до истечения срока аренды, обращаясь к тому серверу, от которого он получил текущие параметры. Если ответа нет или ответ отрицательный, он через некоторое время снова посылает запрос. Так повторяется несколько раз, и если все попытки получить параметры у того же сервера оказываются безуспешными, клиент обращается к другому серверу. Если и другой сервер отвечает отказом, то клиент теряет свои конфигурационные параметры и переходит в режим автономной работы.

Также DHCP-клиент может по своей инициативе досрочно отказаться от выделенных ему параметров.

В сети, где адреса назначаются динамически, нельзя быть уверенным в адресе, который в данный момент имеет тот или иной узел. И такое непостоянство IP-адресов влечет за собой некоторые проблемы.

Во-первых, *возникают сложности при преобразовании символьного доменного имени в IP-адрес*. Действительно, представьте себе функционирование системы DNS, которая должна поддерживать таблицы соответствия символьных имен IP-адресам в условиях, когда последние меняются каждые два часа! Учитывая это обстоятельство, для серверов, к которым

пользователи часто обращаются по символьному имени, назначают статические IP-адреса, оставляя динамические только для клиентских компьютеров. Однако в некоторых сетях количество серверов настолько велико, что их ручное конфигурирование становится слишком обременительным. Это привело к разработке усовершенствованной версии DNS (так называемой динамической системы DNS), в основе которой лежит согласование информационной адресной базы в службах DHCP и DNS.

Во-вторых, *трудно осуществлять удаленное управление и автоматический мониторинг интерфейса* (например, сбор статистики), если в качестве его идентификатора выступает динамически изменяемый IP-адрес.

Наконец, для обеспечения безопасности сети многие сетевые устройства могут блокировать (фильтровать) пакеты, определенные поля которых имеют некоторые заранее заданные значения. Другими словами, при динамическом назначении адресов *усложняется фильтрация пакетов по IP-адресам.*

Последние две проблемы проще всего решаются отказом от динамического назначения адресов для интерфейсов, фигурирующих в системах мониторинга и безопасности.

Выводы

В стеке TCP/IP используются три типа адресов: локальные (называемые также аппаратными), IP-адреса и символьные доменные имена. Все эти типы адресов присваиваются узлам составной сети независимо друг от друга.

IP-адрес имеет длину 4 байта и состоит из номера сети и номера узла. Для определения границы, отделяющей номер сети от номера узла, сегодня используются два подхода. Первый основан на применении классов адресов, второй — масок.

Номер сети назначается централизовано, если сеть является частью Интернета. Назначение IP-адресов узлам сети может происходить либо вручную (администратор сам ведет списки свободных и занятых адресов и конфигурирует сетевой интерфейс), либо автоматически (с использованием протокола DHCP).

Установление соответствия между IP-адресом и аппаратным адресом сетевого интерфейса осуществляется протоколом разрешения адресов (ARP).

В стеке TCP/IP применяется система доменных символьных имен, которая имеет иерархическую древовидную структуру, допускающую использование в имени произвольного количества составных частей. Совокупность имен, у которых несколько старших составных частей совпадают, образуют домен имен. Доменные имена назначаются централизованно, если сеть является частью Интернета, в противном случае — локально.

Соответствие между доменными именами и IP-адресами может устанавливаться как средствами локального хоста с использованием файла hosts, так и с помощью централизованной службы DNS.

Вопросы и задания

1. Какие из адресов могли бы в составной IP-сети являться локальными, а какие нет? Варианты ответов:

   а) адрес VPI/VCI сети ATM;

   б) DNS-адрес X.25, например, w1.120dep;

   в) MAC-адрес, например, 12-B3-3B-51-A2-10;

   г) IP-адрес, например, 113.34.78.01.

2. Какие из следующих утверждений верны всегда?

   а) каждый интерфейс маршрутизатора имеет сетевой адрес;

   б) каждый интерфейс моста/коммутатора имеет сетевой адрес;

   в) каждый маршрутизатор имеет собственный сетевой адрес;

   г) каждый интерфейс маршрутизатора имеет MAC-адрес.

3. Какие из приведенных адресов не могут быть использованы в качестве IP-адресов сетевого интерфейса для узлов Интернета? Для синтаксически правильных адресов определите их класс: A, B, C, D или E. Варианты адресов:

   а) 223.13.123.245; б) 225.0.0.105; в) 194.87.45.0; г) 10.24.255.252;

   д) 125.24.255.255; е) 157.213.255.305; ж) 129.12.255.255; з) 127.0.23.255;

   и) 1.0.0.13; к) 221.1.1.1; л) 192.134.216.255; м) 193.256.254.11.

4. Пусть IP-адрес некоторого узла подсети равен 108.5.18.167, а значение маски для этой подсети — 255.255.240.0. Определите номер подсети. Какое максимальное число сетевых интерфейсов может быть в этой подсети?

5. Пусть вам ничего не известно о структуре сети, но в вашем распоряжении имеется следующая таблица соответствия IP-адресов и DNS-имен нескольких узлов сети.

IP-адрес узла	123.1.0.01	123.1.0.02	123.1.0.03	123.1.0.04	?	?
DNS-имя узла	w1.mgu.ru	w2.mgu.ru	w3.mgu.ru	w4.mgu.ru	w5.mgu.ru	w6.mgu.ru

   Что вы можете сказать об IP-адресах узлов, имеющих DNS-имена w5.mgu.ru и w6.mgu.ru?

6. Пусть вам ничего не известно о структуре сети, но вы знаете DNS-имена некоторых узлов: w1.mgu.ru, w4.mgu.ru и w3.dept.ru. Что вы можете сказать о том, насколько близко территориально находятся они относительно друг друга. Варианты ответов:

   а) узел w1.mgu.ru расположен ближе к w6.mgu.ru, чем к w3.dept.ru;

   б) узел w1.mgu.ru расположен ближе к w3.dept.ru, чем к w6.mgu.ru;

   в) ничего определенного.

7. Сколько ARP-таблиц имеет компьютер? Маршрутизатор?

8. Протокол ARP функционально можно разделить на клиентскую и серверную части. Опишите, какие функции вы отнесли бы к клиентской части, а какие — к серверной?

9. Сколько DHCP-серверов достаточно, чтобы обслужить сеть, разделенную двумя маршрутизаторами?

10. Какое максимальное количество подсетей теоретически можно организовать, если в вашем распоряжении имеется сеть класса B? Какое значение должна при этом иметь маска?

11. В студенческом общежитии живет 200 студентов и каждый из них имеет собственный ноутбук. В общежитии оборудована специальная комната, в которой развернута компьютерная сеть, имеющая 25 коннекторов для подключения компьютеров. Время от времени студенты работают в этом компьютерном классе, подключая свои ноутбуки к сети. Каким количеством IP-адресов должен располагать администратор этой компьютерной сети, чтобы все студенты могли подключаться к сети, не выполняя процедуру конфигурирования своих ноутбуков при каждом посещении компьютерного класса?

ГЛАВА 16 Протокол межсетевого взаимодействия

Эта глава посвящена протоколу **IP** (Intrenet Protocol — межсетевой протокол), описанному в документе RFC 751. В каждой очередной сети, лежащей на пути перемещения пакета, протокол IP обращается к средствам транспортировки этой сети, чтобы с их помощью передать пакет на маршрутизатор, ведущий к следующей сети, или непосредственно на узел-получатель. Таким образом, одной из важнейших функций IP является *поддержание интерфейса с нижележащими технологиями* сетей, образующих составную сеть. Кроме того, в функции протокола IP входит *поддержание интерфейса с протоколами вышележащего транспортного уровня*, в частности с протоколом TCP, который решает все вопросы обеспечения надежной доставки данных по составной сети в стеке TCP/IP.

Протокол IP относится к протоколам *без установления соединений*, он поддерживает обработку каждого IP-пакета как независимой единицы обмена, не связанной с другими пакетами. В протоколе IP нет механизмов, обычно применяемых для обеспечения достоверности конечных данных. Если во время продвижения пакета происходит какая-либо ошибка, то протокол IP по своей инициативе ничего не предпринимает для исправления этой ошибки. Например, если на промежуточном маршрутизаторе пакет был отброшен из-за ошибки по контрольной сумме, то модуль IP не пытается заново послать потерянный пакет. Другими словами, протокол IP реализует политику доставки «по возможности» (с максимальными усилиями).

В этой главе мы подробно рассмотрим основную функцию протокола IP — *маршрутизацию*. Основательно изучим структуру таблиц маршрутизации как без использования, так и с использованием масок. Приведем примеры применения масок одинаковой и переменной длины, перекрывающихся адресных пространств, разделения на подсети и объединения подсетей. Также мы исследуем возможности протокола IP, связанные с фрагментацией пакетов.

Формат IP-пакета

Имеется прямая связь между количеством полей заголовка пакета и функциональной сложностью протокола, который работает с этим заголовком. Чем проще заголовок — тем проще соответствующий протокол. Большая часть действий протокола связана с обработкой той служебной информации, которая переносится в полях заголовка пакета. Изучая назначение каждого поля заголовка IP-пакета, мы получаем не только формальные знания о структуре пакета, но и знакомимся с основными функциями протокола IP.

IP-пакет состоит из полей заголовка и данных. Далее перечислены поля заголовка, показанные на рис. 16.1.

Рис. 16.1. Структура заголовка IP-пакета

Поле **номера версии** занимает 4 бита и идентифицирует версию протокола IP. Сейчас повсеместно используется версия 4 (IPv4), хотя все чаще встречается и новая версия (IPv6).

Значение **длины заголовка** IP-пакета также занимает 4 бита и измеряется в 32-битных словах. Обычно заголовок имеет длину в 20 байт (пять 32-битных слов), но при добавлении некоторой служебной информации это значение может быть увеличено за счет дополнительных байтов в поле параметров. Наибольшая длина заголовка составляет 60 байт.

Поле **типа сервиса** (Type of Service, ToS) имеет и другое, более современное название — байт **дифференцированного обслуживания**, или **DS-байт**. Этим двум названиям соответствуют два варианта интерпретации этого поля. В обоих случаях данное поле служит одной цели — хранению признаков, которые отражают требования к качеству обслуживания пакета. В прежнем варианте первые три бита содержат значение **приоритета** пакета: от самого низкого — 0 до самого высокого — 7. Маршрутизаторы и компьютеры могут принимать во внимание приоритет пакета и обрабатывать более важные пакеты в первую очередь. Следующие три бита поля ToS определяют **критерий выбора маршрута**. Если бит D (Delay — задержка) установлен в 1, то маршрут должен выбираться для минимизации задержки доставки данного пакета, установленный бит T (Throughput — пропускная способность) – для максимизации пропускной способности, а бит R (Reliability — надежность) — для максимизации надежности доставки. Оставшиеся два бита имеют нулевое значение.

Стандарты дифференцированного обслуживания, принятые в конце 90-х годов, дали новое название этому полю и переопределили назначение его битов. В DS-байте также используются только старшие 6 бит, а два младших бита остаются в качестве резерва. Назначение битов DS-байта рассмотрено в разделе «Дифференцированное обслуживание» главы 18.

Поле **общей длины** занимает 2 байта и характеризует общую длину пакета с учетом заголовка и поля данных. Максимальная длина пакета ограничена разрядностью поля, определяющего эту величину, и составляет 65 535 байт, однако в большинстве компьютеров и сетей столь большие пакеты не используются. При передаче по сетям различного типа длина пакета выбирается с учетом максимальной длины пакета протокола нижнего уровня, несущего IP-пакеты. Если это кадры Ethernet, то выбираются пакеты с максимальной длиной 1500 байт, умещающиеся в поле данных кадра Ethernet. В стандартах TCP/IP предусматривается, что все хосты должны быть готовы принимать пакеты длиной вплоть до 576 байт (независимо от того, приходят ли они целиком или фрагментами).

Идентификатор пакета занимает 2 байта и используется для распознавания пакетов, образовавшихся путем деления на части (фрагментации) исходного пакета. Все части (фрагменты) одного пакета должны иметь одинаковое значение этого поля.

Флаги занимают 3 бита и содержат признаки, связанные с фрагментацией. Установленный в 1 бит DF (Do not Fragment — не фрагментировать) запрещает маршрутизатору фрагментировать данный пакет, а установленный в 1 бит MF (More Fragments — больше фрагментов) говорит о том, что данный пакет является промежуточным (не последним) фрагментом. Оставшийся бит зарезервирован.

Поле **смещения фрагмента** занимает 13 бит и задает смещение в байтах поля данных этого фрагмента относительно начала поля данных исходного (нефрагментированного) пакета. Используется при сборке/разборке фрагментов пакетов. Смещение должно быть кратно 8 байт.

Поле **времени жизни** (Time To Live, TTL) занимает один байт и используется для задания предельного срока, в течение которого пакет может перемещаться по сети. Время жизни пакета измеряется в секундах и задается источником. По истечении каждой секунды пребывания на каждом из маршрутизаторов, через которые проходит пакет во время своего «путешествия» по сети, из его текущего времени жизни вычитается единица; единица вычитается и в том случае, если время пребывания было меньше секунды. Поскольку современные маршрутизаторы редко обрабатывают пакет дольше, чем за одну секунду, то время жизни можно интерпретировать как максимальное число транзитных узлов, которые разрешено пройти пакету. Если значение поля времени жизни становится нулевым до того, как пакет достигает получателя, пакет уничтожается. Таким образом, время жизни является своего рода часовым механизмом самоуничтожения пакета.

Поле **протокола верхнего уровня** занимает один байт и содержит идентификатор, указывающий, какому протоколу верхнего уровня принадлежит информация, размещенная в поле данных пакета. Значения идентификаторов для разных протоколов приводятся в документе RFC 1700, доступном по адресу http://www.iana.org. Например, 6 означает, что в пакете находится сообщение протокола TCP, 17 — протокола UDP, 1 — протокола ICMP.

Контрольная сумма заголовка занимает 2 байта (16 бит) и рассчитывается только по заголовку. Поскольку некоторые поля заголовка меняют свое значение в процессе передачи пакета по сети (например, поле времени жизни), контрольная сумма проверяется и повторно рассчитывается на каждом маршрутизаторе и конечном узле как дополнение к сумме всех 16-битных слов заголовка. При вычислении контрольной суммы значение самого поля

контрольной суммы устанавливается в нуль. Если контрольная сумма неверна, то пакет отбрасывается, как только обнаруживается ошибка.

Поля **IP-адресов источника** и **приемника** имеют одинаковую длину — 32 бита.

Поле **параметров** является необязательным и используется обычно только при отладке сети. Это поле состоит из нескольких подполей одного из восьми предопределенных типов. В этих подполях можно указывать точный маршрут, регистрировать проходимые пакетом маршрутизаторы, помещать данные системы безопасности или временные отметки.

Так как число подполей в поле параметров может быть произвольным, то в конце заголовка должно быть добавлено несколько нулевых байтов для **выравнивания** заголовка пакета по 32-битной границе.

Далее приведена распечатка значений полей заголовка одного из реальных IP-пакетов, захваченных в сети Ethernet средствами анализатора протоколов сетевого монитора (Network Monitor, NM) компании Microsoft. В данной распечатке NM в скобках дает шестнадцатеричные значения полей, кроме того, программа иногда представляет числовые коды полей в виде, более удобном для чтения. Например, дружественный программный интерфейс NM интерпретирует код 6 в поле протокола, помещая туда название соответствующего протокола — TCP (см. строку, выделенную полужирным шрифтом).

```
IP: Version = 4 (0x4)
IP: Header Length = 20 (0x14)
IP: Service Type = 0 (0x0)
IP: Precedence = Routine
IP: ...0.... = Normal Delay
IP: ....0... = Normal Throughput
IP: .....0.. = Normal Reliability
IP: Total Length = 54 (0x36)
IP: Identification = 31746 (0x7C02)
IP: Flags Summary = 2 (0x2)
IP: .......0 = Last fragment in datagram
IP: ......1. = Cannot fragment datagram
IP: Fragment Offset = 0 (0x0) bytes
IP: Time to Live = 128 (0x80)
IP: Protocol = TCP - Transmission Control
IP: Checksum = 0xEB86
IP: Source Address = 194.85.135.75
IP: Destination Address = 194.85.135.66
IP: Data: Number of data bytes remaining = 34 (0x0022)
```

Схема IP-маршрутизации

Рассмотрим механизм IP-маршрутизации на примере составной сети, представленной на рис. 16.2. В этой сети 20 маршрутизаторов (изображенных в виде пронумерованных квадрантных блоков) объединяют 18 сетей в общую сеть; N1, N2, ..., N18 — это номера сетей. На каждом маршрутизаторе и конечных узлах *A* и *B* функционируют протоколы IP.

К нескольким интерфейсам (портам) маршрутизаторов присоединяются сети. Каждый интерфейс маршрутизатора можно рассматривать как отдельный узел сети: он имеет сетевой адрес и локальный адрес в той подсети, которая к нему подключена. Например, маршрутизатор под номером 1 имеет три интерфейса, к которым подключены сети N1, N2, N3. На рисунке сетевые адреса этих портов обозначены IP_{11}, IP_{12} и IP_{13}. Интерфейс

IP_{11} является узлом сети N1, и следовательно, в поле номера сети порта IP_{11} содержится номер N1. Аналогично интерфейс IP_{12} — это узел в сети N2, а порт IP_{13} — узел в сети N3. Таким образом, маршрутизатор можно рассматривать как совокупность нескольких узлов, каждый из которых входит в свою сеть. Как единое устройство маршрутизатор не имеет выделенного адреса, ни сетевого, ни локального.

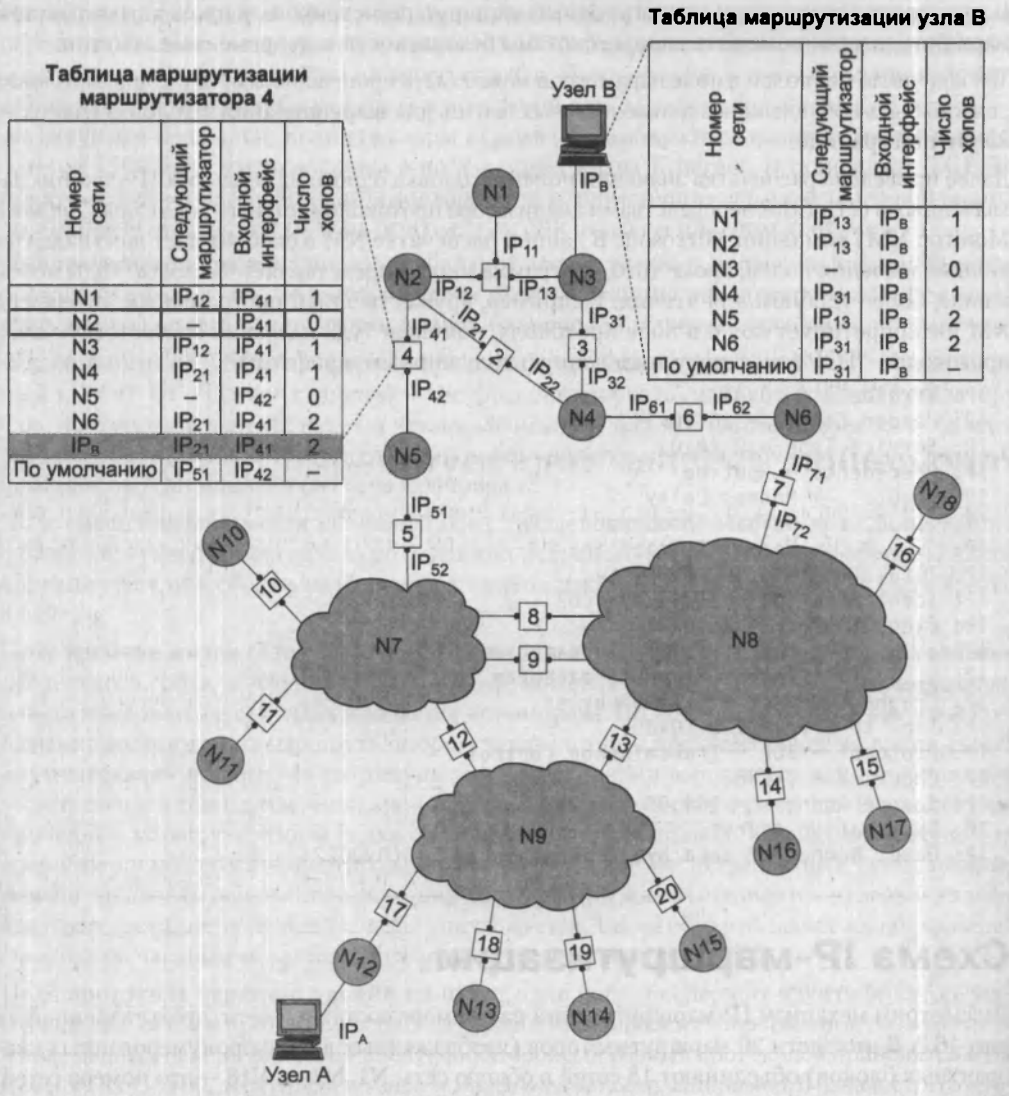


Рис. 16.2. Принципы маршрутизации в составной сети

В сложных составных сетях почти всегда существуют несколько альтернативных маршрутов для передачи пакетов между двумя конечными узлами. Так, пакет, отправленный из узла *A* в узел *B*, может пройти через маршрутизаторы 17, 12, 5, 4 и 1 или маршрутизаторы 17, 13, 7, 6 и 3. Нетрудно найти еще несколько маршрутов между узлами *A* и *B*.

ПРИМЕЧАНИЕ

При наличии у маршрутизатора блока управления (например, по протоколу SNMP) этот блок имеет собственные локальный и сетевой адреса, по которым к нему обращается центральная станция управления. Эти адреса выбираются из того же пула, что и адреса физических интерфейсов маршрутизатора. В технической документации такого рода адреса называются адресами обратной петли (looopback address), или адресами виртуальных интерфейсов (virtual interface address). В отличие от адресов 127.x.x.x, зарезервированных для передачи данных между программными компонентами, находящимися в пределах одного компьютера, адреса виртуальных интерфейсов предполагают обращение к ним извне.

Задачу выбора маршрута из нескольких возможных решают маршрутизаторы, а также конечные узлы. Маршрут выбирается на основании имеющейся у этих устройств информации о текущей конфигурации сети, а также на основании критерия выбора маршрута. В качестве критерия часто выступает задержка прохождения маршрута отдельным пакетом, средняя пропускная способность маршрута для последовательности пакетов или наиболее простой критерий, учитывающий только количество пройденных на маршруте промежуточных маршрутизаторов (*ретрансляционных участков*, или *хопов*). Полученная в результате анализа информация о маршрутах дальнейшего следования пакетов помещается в **таблицу маршрутизации**.

Упрощенная таблица маршрутизации

Используя условные обозначения для сетевых адресов маршрутизаторов и номеров сетей, показанные на рис. 16.2, посмотрим, как могла бы выглядеть таблица маршрутизации, например, в маршрутизаторе 4 (табл. 16.1).

Таблица 16.1. Таблица маршрутизации маршрутизатора 4

Адрес назначения	Сетевой адрес следующего маршрутизатора	Сетевой адрес выходного порта	Расстояние до сети назначения
N1	IP_{12} (R1)	IP_{41}	1
N2	–	IP_{41}	0 (подсоединена)
N3	IP_{12} (R1)	IP_{41}	1
N4	IP_{21} (R2)	IP_{41}	1
N5	–	IP_{42}	0 (подсоединена)
N6	IP_{21} (R2)	IP_{21}	2
IP_B	IP_{21} (R2)	IP_{41}	2
Маршрут по умолчанию	IP_{51} (R5)	IP_{42}	–

ПРИМЕЧАНИЕ

Таблица 16.1 значительно упрощена по сравнению с реальными таблицами, например, здесь отсутствуют столбцы с масками, признаками состояния маршрута, временем, в течение которого действительны записи данной таблицы (их применение будет рассмотрено позже). Вместо номера сети назначения может быть указан полный сетевой адрес отдельного узла назначения. Кроме того, как уже отмечалось, здесь указаны адреса сетей условного формата, не соответствующие какому-либо определенному сетевому протоколу. Тем не менее эта таблица содержит основные поля, имеющиеся в реальных таблицах.

Первый столбец таблицы содержит **адреса назначения пакетов**.

В каждой строке таблицы следом за адресом назначения указывается **сетевой адрес следующего маршрутизатора** (точнее, сетевой адрес интерфейса следующего маршрутизатора), на который надо направить пакет, чтобы тот передвигался по направлению к заданному адресу по рациональному маршруту.

Перед тем как передать пакет следующему маршрутизатору, текущий маршрутизатор должен определить, на какой из нескольких собственных портов (IP_{41} или IP_{42}) он должен поместить данный пакет. Для этого служит третий столбец таблицы маршрутизации, содержащий **сетевые адреса выходных интерфейсов**.

Некоторые реализации сетевых протоколов допускают наличие в таблице маршрутизации сразу *нескольких строк*, соответствующих одному и тому же адресу назначения. В этом случае при выборе маршрута принимается во внимание столбец, представляющий расстояние до сети назначения. При этом расстояние измеряется в любой метрике, используемой в соответствии с заданным в сетевом пакете критерием. Расстояние может измеряться временем прохождения пакета по линиям связи, различными характеристиками надежности линий связи на данном маршруте, пропускной способностью или другой величиной, отражающей качество данного маршрута по отношению к заданному критерию. В табл. 16.1 расстояние между сетями измеряется хопами. Расстояние для сетей, непосредственно подключенных к портам маршрутизатора, здесь принимается равным 0, однако в некоторых реализациях отсчет расстояний начинается с 1.

Когда пакет поступает на маршрутизатор, модуль IP извлекает из его заголовка номер сети назначения и последовательно сравнивает его с номерами сетей из каждой строки таблицы. Строка с совпавшим номером сети показывает ближайший маршрутизатор, на который следует направить пакет. Например, если на какой-либо порт маршрутизатора 4 поступает пакет, адресованный в сеть N6, то из таблицы маршрутизации следует, что адрес следующего маршрутизатора — IP_{21}, то есть очередным этапом движения данного пакета будет движение к порту 1 маршрутизатора 2.

Чаще всего в качестве адреса назначения в таблице указывается не весь IP-адрес, а только номер сети назначения. Таким образом, для всех пакетов, направляемых в одну и ту же сеть, протокол IP будет предлагать один и тот же маршрут (мы пока не принимаем во внимание возможные изменения состояния сети, такие как отказы маршрутизаторов или обрывы кабелей). Однако в некоторых случаях возникает необходимость для одного из узлов сети определить **специфический маршрут**, отличающийся от маршрута, заданного для всех остальных узлов сети. Для этого в таблицу маршрутизации помещают для данного узла отдельную строку, содержащую его полный IP-адрес и соответствующую маршрутную информацию. Такого рода запись имеется в табл. 16.1 для узла *B*. Пусть, например, администратор маршрутизатора 4, руководствуясь соображениями безопасности, решил, что пакеты, следующие в узел *B* (полный адрес IP_B), должны идти через маршрутизатор 2 (интерфейс IP_{21}), а не маршрутизатор 1 (интерфейс IP_{12}), через который передаются пакеты всем остальным узлам сети N3. Если в таблице имеются записи о маршрутах как к сети в целом, так и к ее отдельному узлу, то при поступлении пакета, адресованного данному узлу, маршрутизатор отдаст предпочтение специфическому маршруту.

Поскольку пакет может быть адресован *в любую сеть* составной сети, может показаться, что каждая таблица маршрутизации должна иметь записи обо *всех* сетях, входящих в составную сеть. Однако при таком подходе в случае крупной сети объем таблиц маршрутизации может оказаться очень большим, что повлияет на время ее просмотра, потребует

много места для хранения и т. п. Поэтому на практике широко известен прием уменьшения количества записей в таблице маршрутизации, основанный на введении **маршрута по умолчанию** (default route), учитывающего особенности топологии сети. Рассмотрим, например, маршрутизаторы, находящиеся на периферии составной сети. В их таблицах достаточно записать номера только тех сетей, которые непосредственно подсоединены к данному маршрутизатору или расположены поблизости на тупиковых маршрутах. Обо всех же остальных сетях можно сделать в таблице единственную запись, указывающую на маршрутизатор, через который пролегает путь ко всем этим сетям. Такой маршрутизатор называется **маршрутизатором по умолчанию** (default router). В нашем примере на маршрутизаторе 4 имеются специфические маршруты только для пакетов, следующих в сети N1–N6. Для всех остальных пакетов, адресованных в сети N7–N18, маршрутизатор предлагает продолжить путь через один и тот же порт IP_{51} маршрутизатора 5, который в данном случае и является маршрутизатором по умолчанию.

Таблицы маршрутизации конечных узлов

Задачу маршрутизации решают не только промежуточные узлы (маршрутизаторы), но и конечные узлы — компьютеры. Решение этой задачи начинается с того, что средствами протокола IP на конечном узле определяется, направлен ли пакет в другую сеть или адресован какому-нибудь узлу данной сети. Если номер сети назначения совпадает с номером данной сети, это означает, что пакет маршрутизировать не требуется. В противном случае маршрутизация нужна.

Структуры таблиц маршрутизации конечных узлов и транзитных маршрутизаторов аналогичны. Обратимся снова к сети, изображенной на рис. 16.2. Таблица маршрутизации конечного узла *B*, принадлежащего сети N3, могла бы выглядеть так, как табл. 16.2. Здесь IP_B — сетевой адрес интерфейса компьютера *B*. На основании этой таблицы конечный узел *B* выбирает, на какой из двух имеющихся в локальной сети N3 маршрутизаторов (R1 или R3) следует посылать тот или иной пакет.

Таблица 16.2. Таблица маршрутизации конечного узла B

Номер сети назначения	Сетевой адрес следующего маршрутизатора	Сетевой адрес выходного порта	Расстояние до сети назначения
N1	IP_{13} (R1)	IP_B	1
N2	IP_{13} (R1)	IP_B	1
N3	—	IP_B	0
N4	IP_{31} (R3)	IP_B	1
N5	IP_{13} (R1)	IP_B	2
N6	IP_{31} (R3)	IP_B	2
Маршрут по умолчанию	IP_{31} (R3)	IP_B	—

Конечные узлы в еще большей степени, чем маршрутизаторы, пользуются приемом маршрутизации по умолчанию. Хотя они также в общем случае имеют в своем распоряжении таблицу маршрутизации, ее объем обычно незначителен, что объясняется периферийным расположением всех конечных узлов. Конечный узел часто вообще работает без таблицы

маршрутизации, имея только сведения об адресе маршрутизатора по умолчанию. При наличии одного маршрутизатора в локальной сети этот вариант — единственно возможный для всех конечных узлов. Но даже при наличии нескольких маршрутизаторов в локальной сети, когда перед конечным узлом стоит проблема их выбора, часто в компьютерах для повышения производительности прибегают к заданию маршрута по умолчанию.

Рассмотрим таблицу маршрутизации другого конечного узла составной сети — узла *A* (табл. 16.3). Компактный вид таблицы маршрутизации узла *A* отражает тот факт, что все пакеты, направляемые из узла *A*, либо не выходят за пределы сети N12, либо непременно проходят через порт 1 маршрутизатора 17. Этот маршрутизатор и определен в таблице маршрутизации в качестве маршрутизатора по умолчанию.

Таблица 16.3. Таблица маршрутизации конечного узла A

Номер сети назначения	Сетевой адрес следующего маршрутизатора	Сетевой адрес выходного порта	Расстояние до сети назначения
N12	—	IP_A	0
Маршрут по умолчанию	$IP_{17,1}$ (R17)	IP_A	—

Еще одним отличием работы маршрутизатора и конечного узла является способ построения таблицы маршрутизации. Если маршрутизаторы, как правило, автоматически создают таблицы маршрутизации, обмениваясь служебной информацией, то для конечных узлов таблицы маршрутизации часто создаются вручную администраторами и хранятся в виде постоянных файлов на дисках.

Просмотр таблиц маршрутизации без масок

Рассмотрим алгоритм просмотра таблицы маршрутизации, реализуемый на маршрутизаторе протоколом IP. При его описании мы будем использовать табл. 16.1 и рис. 16.2.

1. Пусть на один из интерфейсов маршрутизатора поступает пакет. Протокол IP извлекает из пакета IP-адрес назначения (предположим, адрес назначения IP_B).
2. Выполняется *первая фаза* просмотра таблицы — *поиск конкретного маршрута к узлу*. IP-адрес (целиком) последовательно строка за строкой сравнивается с содержимым поля адреса назначения таблицы маршрутизации. Если произошло совпадение (как в табл. 16.1), то из соответствующей строки извлекаются адрес следующего маршрутизатора (IP_{21}) и идентификатор выходного интерфейса (IP_{41}). На этом просмотр таблицы заканчивается.
3. Предположим теперь, что в таблице нет строки с адресом назначения IP_B, а значит, совпадения не произошло. В этом случае протокол IP переходит ко *второй фазе* просмотра — *поиску маршрута к сети назначения*. Из IP-адреса выделяется номер сети (в нашем примере из адреса IP_B выделяется номер сети N3), и таблица снова просматривается на предмет совпадения номера сети в какой-либо строке с номером сети из пакета. При совпадении (в нашем примере оно произошло) из соответствующей строки таблицы извлекаются адрес следующего маршрутизатора (IP_{12}) и идентификатор выходного интерфейса (IP_{41}). Просмотр таблицы на этом завершается.
4. Наконец, предположим, что адрес назначения в пакете был таков, что совпадения не произошло ни в первой, ни во второй фазах просмотра. В таком случае средствами про-

токола IP либо выбирается маршрут по умолчанию (и пакет направляется по адресу IP_{51}), либо, если маршрут по умолчанию отсутствует, пакет отбрасывается[1]. Просмотр таблицы на этом заканчивается.

ВНИМАНИЕ

Последовательность фаз в данном алгоритме строго определена, в то время как последовательность просмотра или, что одно и то же, порядок расположения строк в таблице, включая запись о маршруте по умолчанию, никак не сказывается на результате.

Примеры таблиц маршрутизации разных форматов

Структура реальных таблиц маршрутизации стека TCP/IP в целом соответствует упрощенной структуре рассмотренных ранее таблиц. Отметим, однако, что вид таблицы IP-маршрутизации зависит от конкретной реализации стека TCP/IP. Приведем пример нескольких вариантов таблицы маршрутизации, с которыми мог бы работать маршрутизатор R1 в сети, представленной на рис. 16.3.

Начнем с «придуманного» предельно упрощенного варианта таблицы маршрутизации (табл. 16.4). Здесь имеются три маршрута к сетям (записи 56.0.0.0, 116.0.0.0 и 129.13.0.0), две записи о непосредственно подсоединенных сетях (198.21.17.0 и 213.34.12.0), а также запись о маршруте по умолчанию.

Таблица 16.4. Упрощенная таблица маршрутизации маршрутизатора R1

Адрес сети назначения	Адрес следующего маршрутизатора	Адрес выходного интерфейса	Расстояние до сети назначения
56.0.0.0	213.34.12.4	213.34.12.3	15
116.0.0.0	213.34.12.4	213.34.12.3	13
129.13.0.0	198.21.17.6	198.21.17.5	2
198.21.17.0	198.21.17.5	198.21.17.5	1 (подсоединена)
213.34.12.0	213.34.12.3	213.34.12.3	1 (подсоединена)
Маршрут по умолчанию	198.21.17.7	198.21.17.5	—

Более сложный вид имеют таблицы, которые генерируются в промышленно выпускаемом сетевом оборудовании.

Если представить, что в качестве маршрутизатора R1 в данной сети работает штатный программный маршрутизатор операционной системы Microsoft Windows XP, то его таблица маршрутизации могла бы выглядеть так, как табл. 16.5.

[1] Стандарты технологии TCP/IP не требуют, чтобы в таблице маршрутизации непременно содержались маршруты для всех пакетов, которые могут прийти на его интерфейсы, более того, в таблице может отсутствовать маршрут по умолчанию.

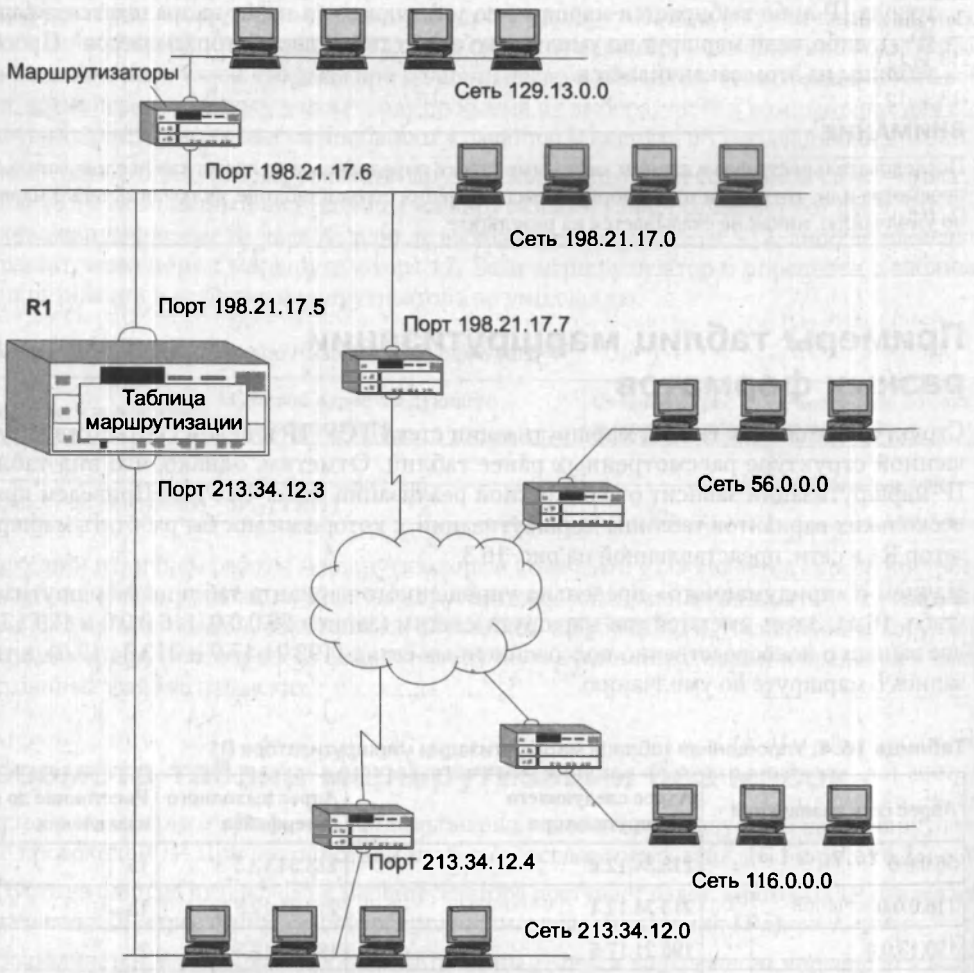


Рис. 16.3. Пример маршрутизируемой сети

Таблица 16.5. Таблица программного маршрутизатора ОС Windows XP

Сетевой адрес	Маска	Адрес шлюза	Интерфейс	Метрика
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
0.0.0.0	0.0.0.0	198.21.17.7	198.21.17.5	1
56.0.0.0	255.0.0.0	213.34.12.4	213.34.12.3	15
116.0.0.0	255.0.0.0	213.34.12.4	213.34.12.3	13
129.13.0.0	255.255.0.0	198.21.17.6	198.21.17.5	2
198.21.17.0	255.255.255.0	198.21.17.5	198.21.17.5	1
198.21.17.5	255.255.255.255	127.0.0.1	127.0.0.1	1
198.21.17.255	255.255.255.255	198.21.17.5	198.21.17.5	1

Сетевой адрес	Маска	Адрес шлюза	Интерфейс	Метрика
213.34.12.0	255.255.255.0	213.34.12.3	213.34.12.3	1
213.34.12.3	255.255.255.255	127.0.0.1	127.0.0.1	1
213.34.12.255	255.255.255.255	213.34.12.3	213.34.12.3	1
224.0.0.0	224.0.0.0	198.21.17.6	198.21.17.6	1
224.0.0.0	224.0.0.0	213.34.12.3	213.34.12.3	1
255.255.255.255	255.255.255.255	198.21.17.6	198.21.17.6	1

Если на месте маршрутизатора R1 установить один из популярных *аппаратных* маршрутизаторов, то его таблица маршрутизации для этой же сети может выглядеть совсем иначе (табл. 16.6).

Таблица 16.6. Таблица маршрутизации аппаратного маршрутизатора

Адрес назначения	Маска	Шлюз	Метрика	Статус	TTL	Источник
198.21.17.0	255.255.255.0	198.21.17.5	0	Up	—	Подключена
213.34.12.0	255.255.255.0	213.34.12.3	0	Up	—	Подключена
56.0.0.0	255.0.0.0	213.34.12.4	14	Up	—	Статическая
116.0.0.0	255.0.0.0	213.34.12.4	12	Up	—	Статическая
129.13.0.0	255.255.0.0	198.21.17.6	1	Up	160	RIP

И наконец табл. 16.7 представляет собой таблицу маршрутизации для того же маршрутизатора R1, реализованного в виде программного маршрутизатора одной из версий операционной системы Unix.

Таблица 16.7. Таблица маршрутизации Unix-маршрутизатора

Адрес назначения	Шлюз	Флаги	Число ссылок	Загрузка	Интерфейс
127.0.0.0	127.0.0.1	UH	1	154	lo0
Маршрут по умолчанию	198.21.17.7	UG	5	43270	le0
198.21.17.0	198.21.17.5	U	35	246876	le0
213.34.12.0	213.34.12.3	U	44	132435	le1
129.13.0.0	198.21.1.7.6	UG	6	16450	le0
56.0.0.0	213.34.12.4	UG	12	5764	le1
116.0.0.0	213.34.12.4	UG	21	23544	le1

ПРИМЕЧАНИЕ

Заметим, что поскольку между структурой сети и таблицей маршрутизации нет однозначного соответствия, для каждого из приведенных вариантов таблицы можно предложить свои «подварианты», отличающиеся выбранным маршрутом к той или иной сети. В данном случае внимание концентрируется на существенных различиях в форме представления маршрутной информации разными реализациями маршрутизаторов.

Несмотря на достаточно заметные внешние различия, во всех трех «реальных» таблицах присутствуют все ключевые данные из рассмотренной упрощенной таблицы, без которых невозможна маршрутизация пакетов.

К таким данным, во-первых, относятся *адреса сети назначения* (столбцы «Адрес назначения» в аппаратном маршрутизаторе и маршрутизаторе Unix или столбец «Сетевой адрес» в маршрутизаторе ОС Windows XP).

Вторым обязательным полем таблицы маршрутизации является *адрес следующего маршрутизатора* (столбцы «Шлюз» в аппаратном маршрутизаторе и маршрутизаторе Unix или столбец «Адрес шлюза» в маршрутизаторе ОС Windows XP).

Третий ключевой параметр — *адрес порта*, на который нужно направить пакет, в некоторых таблицах указывается прямо (столбец «Интерфейс» в таблице маршрутизатора ОС Windows XP), а в некоторых — косвенно. Так, в таблице маршрутизатора Unix вместо адреса порта задается его условное наименование — le0 для порта с адресом 198.21.17.5, le1 для порта с адресом 213.34.12.3 и lo0 для внутреннего порта с адресом 127.0.0.1. В аппаратном маршрутизаторе поле, обозначающее выходной порт в какой-либо форме, вообще отсутствует. Это объясняется тем, что адрес выходного порта всегда можно косвенно определить по адресу следующего маршрутизатора. Например, определим по табл. 16.6 адрес выходного порта для сети 56.0.0.0. Из таблицы следует, что следующим маршрутизатором для этой сети будет маршрутизатор с адресом 213.34.12.4. Адрес следующего маршрутизатора должен принадлежать одной из непосредственно присоединенных к маршрутизатору сетей, и в данном случае это сеть 213.34.12.0. Маршрутизатор имеет порт, присоединенный к этой сети, и адрес этого порта 213.34.12.3 мы находим в столбце «Шлюз» второй строки таблицы маршрутизации, которая описывает непосредственно присоединенную сеть 213.34.12.0. Для непосредственно присоединенных сетей адресом следующего маршрутизатора всегда является адрес собственного порта маршрутизатора. Таким образом, для сети 56.0.0 адресом выходного порта является 213.34.12.3.

Стандартным решением сегодня является использование поля маски в каждой записи таблицы, как это сделано в таблицах маршрутизатора ОС Windows XP и аппаратного маршрутизатора (столбцы «Маска»). Механизм обработки масок при принятии решения маршрутизаторами рассматривается далее. Отсутствие поля маски говорит о том, что либо маршрутизатор рассчитан на работу только с тремя стандартными классами адресов, либо для всех записей используется одна и та же маска, что снижает гибкость маршрутизации.

Поскольку в таблице маршрутизации маршрутизатора Unix каждая сеть назначения упомянута только один раз, а значит, возможность выбора маршрута отсутствует, то поле метрики является необязательным параметром. В остальных двух таблицах поле метрики используется только для указания на то, что сеть подключена непосредственно. Метрика 0 для аппаратного маршрутизатора или 1 для маршрутизатора ОС Windows XP говорит маршрутизатору, что эта сеть непосредственно подключена к его порту, а другое значение метрики соответствует удаленной сети. Выбор метрики для непосредственно подключенной сети (1 или 0) является произвольным, главное, чтобы метрика удаленной сети отсчитывалась с учетом этого выбранного начального значения. В маршрутизаторе Unix используется поле признаков, где флаг G (Gateway — шлюз) отмечает удаленную сеть, а его отсутствие — непосредственно подключенную.

Признак непосредственно подключенной сети говорит маршрутизатору, что пакет уже достиг своей сети, поэтому протокол IP активизирует ARP-запрос относительно IP-адреса *узла назначения*, а не следующего маршрутизатора.

Однако существуют ситуации, когда маршрутизатор должен обязательно хранить значение метрики для записи о каждой удаленной сети. Эти ситуации возникают, когда записи в таблице маршрутизации являются результатом работы некоторых протоколов маршрутизации, например протокола RIP. В таких протоколах новая информация о какой-либо удаленной сети сравнивается с информацией, содержащейся в таблице в данный момент, и если значение новой метрики лучше текущей, то новая запись вытесняет имеющуюся. В таблице маршрутизатора Unix поле метрики отсутствует, и это значит, что он не использует протокол RIP.

Флаги записей присутствуют только в таблице маршрутизатора Unix.

- U — маршрут активен и работоспособен. Аналогичный смысл имеет поле статуса в аппаратном маршрутизаторе.
- H — признак специфического маршрута к определенному хосту.
- G — означает, что маршрут пакета проходит через промежуточный маршрутизатор (шлюз). Отсутствие этого флага отмечает непосредственно подключенную сеть.
- D — означает, что маршрут получен из перенаправленного сообщения протокола ICMP. Этот признак может присутствовать только в таблице маршрутизации *конечного узла*. Признак означает, что конечный узел при какой-то предыдущей передаче пакета выбрал не самый рациональный следующий маршрутизатор на пути к данной сети, и этот маршрутизатор с помощью протокола ICMP сообщил конечному узлу, что все последующие пакеты к данной сети нужно отправлять через другой маршрутизатор.

В таблице маршрутизатора Unix используются еще два поля, имеющих справочное значение. Поле числа ссылок показывает, сколько раз на данный маршрут ссылались при продвижении пакетов. Поле загрузки отражает количество байтов, переданных по данному маршруту.

В записях таблиц аппаратного маршрутизатора также имеются два справочных поля. Поле **времени жизни записи** (TTL) в данном случае никак не связано со временем жизни пакета. Здесь оно показывает время, в течение которого значение данной записи еще действительно. **Поле источника** говорит об источнике появления записи в таблице маршрутизации.

Источники и типы записей в таблице маршрутизации

Практически для всех маршрутизаторов существуют *три* основных источника записей в таблице.

- Одним из источников записей в таблице маршрутизации является **программное обеспечение стека TCP/IP**, которое при инициализации маршрутизатора автоматически заносит в таблицу несколько записей, в результате чего создается так называемая минимальная таблица маршрутизации. Программное обеспечение формирует записи о *непосредственно подключенных сетях* и маршрутах по умолчанию, информация о которых появляется в стеке при ручном конфигурировании интерфейсов компьютера или маршрутизатора. К таким записям в приведенных примерах относятся записи о сетях 213.34.12.0 и 198.21.17.0, а также запись о маршруте по умолчанию в маршрутизаторе Unix и запись 0.0.0.0 в маршрутизаторе ОС Windows XP. Кроме того, программное обеспечение автоматически заносит в таблицу маршрутизации записи *об адресах особого назначения*. В приведенных примерах таблица маршрутизатора ОС Windows 2000 содержит наиболее полный набор записей такого рода. Несколько записей в этой таблице

связано с особым адресом 127.0.0.0. Записи с адресом 224.0.0.0 требуются для обработки групповых адресов. Кроме того, в таблицу могут быть занесены адреса, предназначенные для обработки широковещательных рассылок (например, записи 8 и 11 содержат адрес отправки широковещательного сообщения в соответствующих подсетях, а последняя запись в таблице — адрес ограниченной широковещательной рассылки). Заметим, что в некоторых таблицах записи об особых адресах вообще отсутствуют.

- Еще одним источником записей в таблице является **администратор**, непосредственно формирующий записи с помощью некоторой системной утилиты, например программы route, имеющейся в операционных системах Unix и Windows XP. В аппаратных маршрутизаторах также всегда имеется команда для ручного задания записей таблицы маршрутизации. Заданные вручную записи всегда являются *статическими*, то есть они не имеют срока жизни. Эти записи могут быть как постоянными, то есть сохраняющимися при перезагрузке маршрутизатора, так и временными, хранящимися в таблице только до выключения устройства. Часто администратор вручную заносит запись о маршруте по умолчанию. Таким же образом в таблицу маршрутизации может быть внесена запись о специфическом для узла маршруте.
- И наконец, третьим источником записей могут быть **протоколы маршрутизации**, такие как RIP или OSPF. Эти записи всегда являются *динамическими*, то есть имеют ограниченный срок жизни.

Программные маршрутизаторы Windows XP и Unix не показывают источник появления той или иной записи в таблице, а аппаратный маршрутизатор использует для этой цели поле источника. В приведенном в табл. 16.6 примере первые две записи созданы программным обеспечением стека на основании данных о конфигурации портов маршрутизатора — это показывает признак «Подключена». Следующие две записи обозначены как статические — это означает, что их ввел вручную администратор. Последняя запись является следствием работы протокола RIP, поэтому в ее поле «TTL» имеется значение 160.

Пример IP-маршрутизации без масок

Рассмотрим процесс продвижения пакета в составной сети на примере IP-сети, показанной на рис. 16.4. При этом будем считать, что все узлы сети, рассматриваемой в примере, имеют *адреса, основанные на классах*. Особое внимание будет уделено взаимодействию протокола IP с протоколами разрешения адресов ARP и DNS.

Итак, пусть пользователю компьютера cit.mgu.com, находящегося в сети 129.13.0.0, необходимо установить связь с FTP-сервером. Пользователю известно символьное имя сервера unix.mgu.com, поэтому он набирает на клавиатуре команду обращения к FTP-серверу по имени:

```
> ftp unix.mgu.com
```

Выполнение этой команды инициирует три последовательные операции:

1. DNS-клиент (работающий на компьютере cit.mgu.com) передает DNS-серверу сообщение, в котором содержится запрос об IP-адресе сервера unix.mgu.com, с которым он хочет связаться по протоколу FTP.
2. DNS-сервер, выполнив поиск, передает ответ DNS-клиенту о найденном IP-адресе сервера unix.mgu.com.
3. FTP-клиент (работающий на том же компьютере cit.mgu.com), используя найденный IP-адрес сервера unix.mgu.com, передает сообщение работающему на нем FTP-серверу.

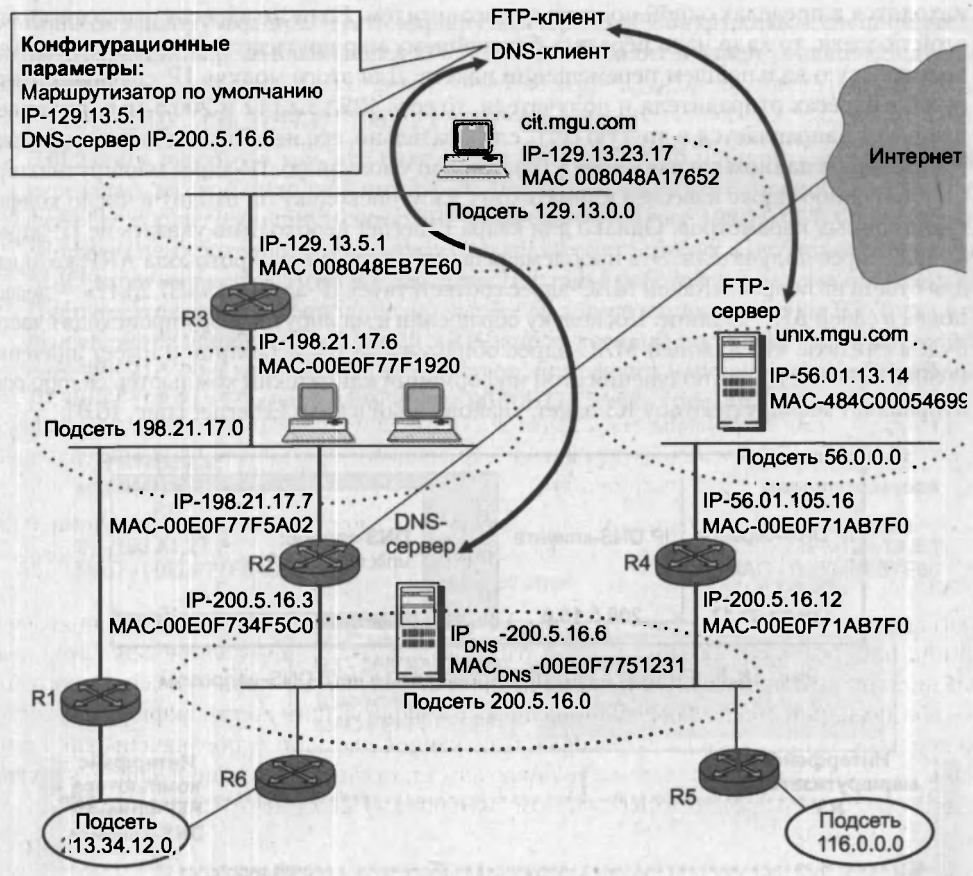


Рис. 16.4. Пример IP-маршрутизации

Давайте последовательно, по шагам, рассмотрим, как при решении этих задач взаимодействуют между собой протоколы DNS, IP, ARP и Ethernet и что происходит при этом с кадрами и пакетами.

1. *Формирование IP-пакета с инкапсулированным в него DNS-запросом.* Программный модуль FTP-клиента, получив команду `> ftp unix.mgu.com`, передает запрос к работающей на этом же компьютере клиентской части протокола DNS, которая, в свою очередь, формирует к DNS-серверу запрос, интерпретируемый примерно так: «Какой IP-адрес соответствует символьному имени unix.mgu.com?» Запрос упаковывается в UDP-дейтаграмму, затем в IP-пакет. В заголовке пакета в качестве адреса назначения указывается IP-адрес 200.5.16.6 DNS-сервера. Этот адрес известен программному обеспечению клиентского компьютера, так как он входит в число его конфигурационных параметров. Сформированный IP-пакет будет перемещаться по сети в неизменном виде (как показано на рис. 16.5), пока не дойдет до адресата — DNS-сервера.
2. *Передача кадра Ethernet с IP-пакетом маршрутизатору R3.* Для передачи этого IP-пакета необходимо его упаковать в кадр Ethernet, указав в заголовке MAC-адрес получателя. Технология Ethernet способна доставлять кадры только тем адресатам, которые

находятся в пределах одной подсети с отправителем. Если же адресат расположен вне этой подсети, то кадр надо передать ближайшему маршрутизатору, чтобы тот взял на себя заботу о дальнейшем перемещении пакета. Для этого модуль IP, сравнив номера сетей в адресах отправителя и получателя, то есть 129.13.23.17 и 200.5.16.6, выясняет, что пакет направляется в другую сеть, следовательно, его необходимо передать маршрутизатору, в данном случае маршрутизатору по умолчанию. IP-адрес маршрутизатора по умолчанию также известен клиентскому узлу, поскольку он входит в число конфигурационных параметров. Однако для кадра Ethernet необходимо указать не IP-адрес, а MAC-адрес получателя. Эта проблема решается с помощью протокола ARP, который для ответа на вопрос: «Какой MAC-адрес соответствует IP-адресу 194.87.23.1?» — делает поиск в своей ARP-таблице. Поскольку обращения к маршрутизатору происходят часто, будем считать, что нужный MAC-адрес обнаруживается в таблице и имеет значение 008048EB7E60. После получения этой информации клиентский компьютер cit.mgu.com отправляет маршрутизатору R3 пакет, упакованный в кадр Ethernet (рис. 16.6).

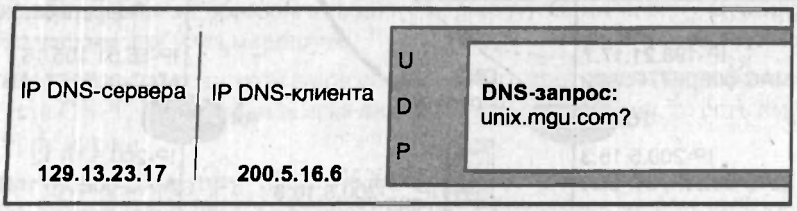


Рис. 16.5. IP-пакет с инкапсулированным в него DNS-запросом

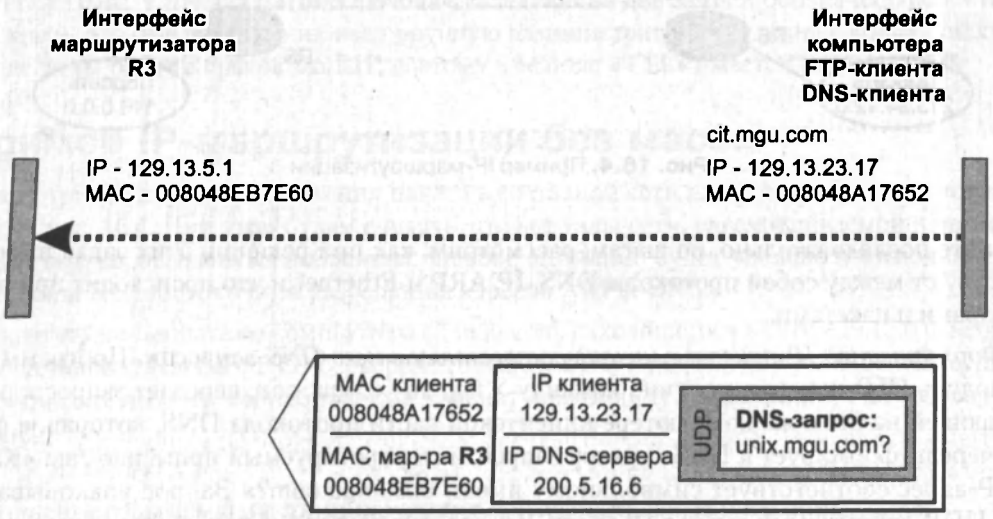


Рис. 16.6. Кадр Ethernet с инкапсулированным IP-пакетом, отправленный с клиентского компьютера

3. *Определение IP-адреса и MAC-адреса следующего маршрутизатора R2.* Кадр принимается интерфейсом 129.13.5.1 маршрутизатора R3. Протокол Ethernet, работающий на этом интерфейсе, извлекает из этого кадра IP-пакет и передает его протоколу IP. Протокол IP находит в заголовке пакета адрес назначения 200.5.16.6 и просматривает записи своей

таблицы маршрутизации. Пусть маршрутизатор R3 не обнаруживает специфического маршрута для адреса назначения 200.5.16.6, но находит в своей таблице следующую запись:

200.5.16.0 198.21.17.7 198.21.17.6

Эта запись говорит о том, что пакеты для сети 200.5.16.0 маршрутизатор R3 должен передавать на свой выходной интерфейс 198.21.17.6, с которого они поступят на интерфейс следующего маршрутизатора R2, имеющего IP-адрес 198.21.17.7. Однако знания IP-адреса недостаточно, чтобы передать пакет по сети Ethernet. Необходимо определить MAC-адрес маршрутизатора R3. Как известно, такой работой занимается протокол ARP. Пусть на этот раз в ARP-таблице нет записи об адресе маршрутизатора R3. Тогда в сеть отправляется широковещательный ARP-запрос, который поступает на все интерфейсы сети 198.21.17.0. Ответ приходит только от интерфейса маршрутизатора R3: «Я имею IP-адрес 198.21.17.7 и мой MAC-адрес 00E0F77F5A02» (рис. 16.7).

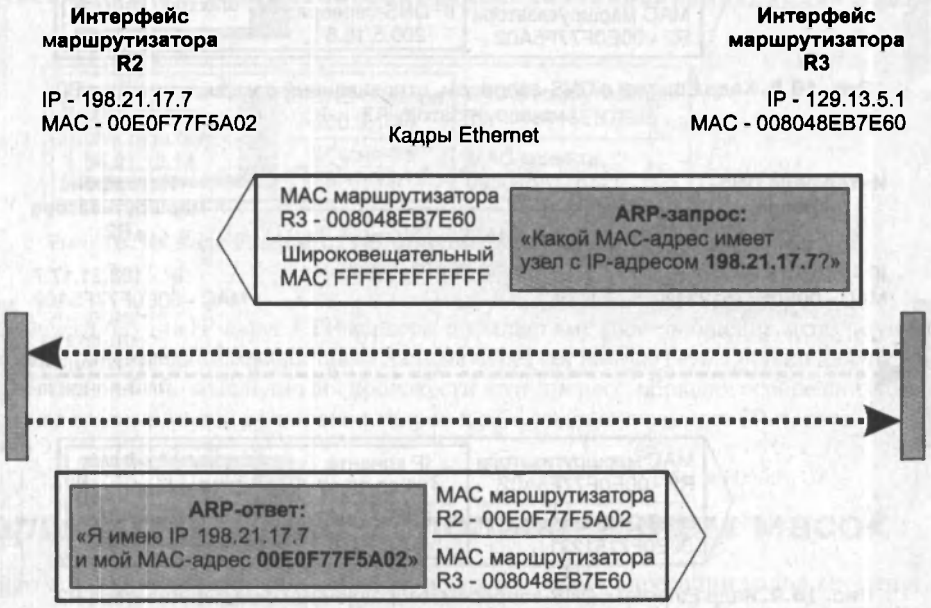


Рис. 16.7. Кадры Ethernet с инкапсулированными ARP-запросом и ARP-ответом

Теперь, зная MAC-адрес маршрутизатора R2 (00E0F77F5A02), маршрутизатор R3 отсылает ему IP-пакет с DNS-запросом (рис. 16.8).

4. *Маршрутизатор R2 доставляет пакет DNS-серверу.* Модуль IP на маршрутизаторе R2 действует в соответствии с уже не раз описанной нами процедурой: отбросив заголовок кадра Ethernet, он извлекает из пакета IP-адрес назначения и просматривает свою таблицу маршрутизации. Там он обнаруживает, что сеть назначения 200.5.16.0 является непосредственно присоединенной к его второму интерфейсу. Следовательно, пакет не нужно маршрутизировать, однако требуется определить MAC-адрес узла назначения. Протокол ARP «по просьбе» протокола IP находит (либо из ARP-таблицы, либо по запросу) требуемый MAC-адрес 00E0F7751231 DNS-сервера. Получив ответ

о MAC-адресе, маршрутизатор R2 отправляет в сеть назначения кадр Ethernet с DNS-запросом (рис. 16.9).

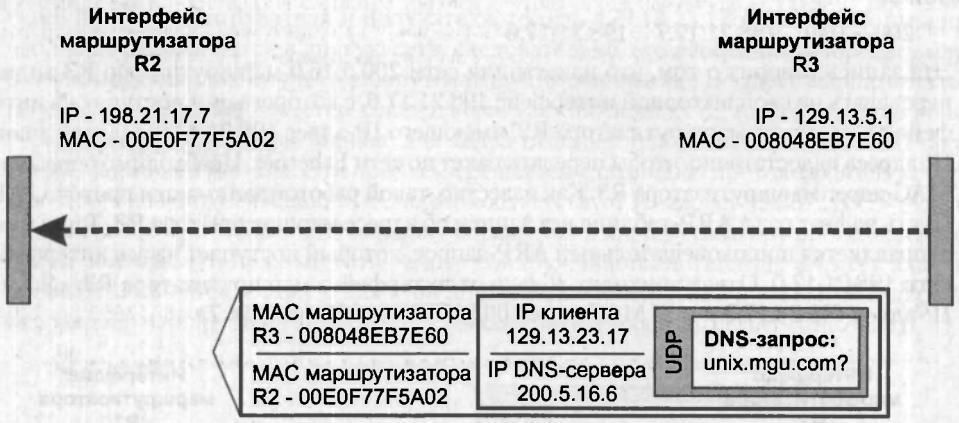


Рис. 16.8. Кадр Ethernet с DNS-запросом, отправленный с маршрутизатора R3 маршрутизатору R2

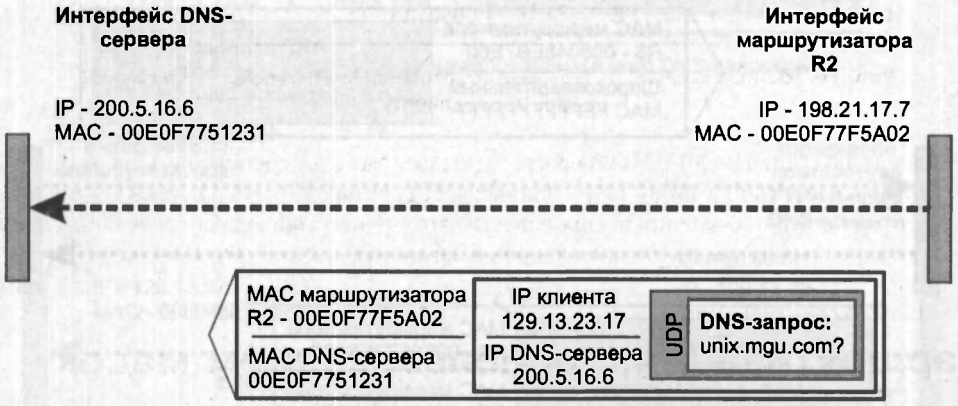


Рис. 16.9. Кадр Ethernet с DNS-запросом, отправленный с маршрутизатора R2

5. *Сетевой адаптер DNS-сервера захватывает кадр Ethernet*, обнаруживает совпадение MAC-адреса назначения, содержащегося в заголовке, со своим собственным адресом и направляет его модулю IP. После анализа полей заголовка IP из пакета извлекаются данные вышележащих протоколов. DNS-запрос передается программному модулю DNS-сервера DNS-сервер просматривает свои таблицы, возможно, обращается к другим DNS-серверам и в результате формирует ответ, смысл которого состоит в следующем: «Символьному имени unix.mgu.com соответствует IP-адрес 56.01.13.14».

Процесс доставки DNS-ответа клиенту cit.mgu.com совершенно аналогичен процессу передачи DNS-запроса, который мы только что так подробно описали. Работая в тесной кооперации, протоколы IP, ARP и Ethernet передают клиенту DNS-ответ через всю составную сеть (рис. 16.10).

ПРИМЕЧАНИЕ

Заметим, что во время всего путешествия пакета по составной сети от клиентского компьютера до DNS-сервера IP-адреса получателя и отправителя в полях заголовка IP-пакета не изменяются. Зато в заголовке каждого нового кадра, который переносил пакет от одного маршрутизатора к другому, MAC-адреса отправителя и получателя изменяются на каждом отрезке пути.

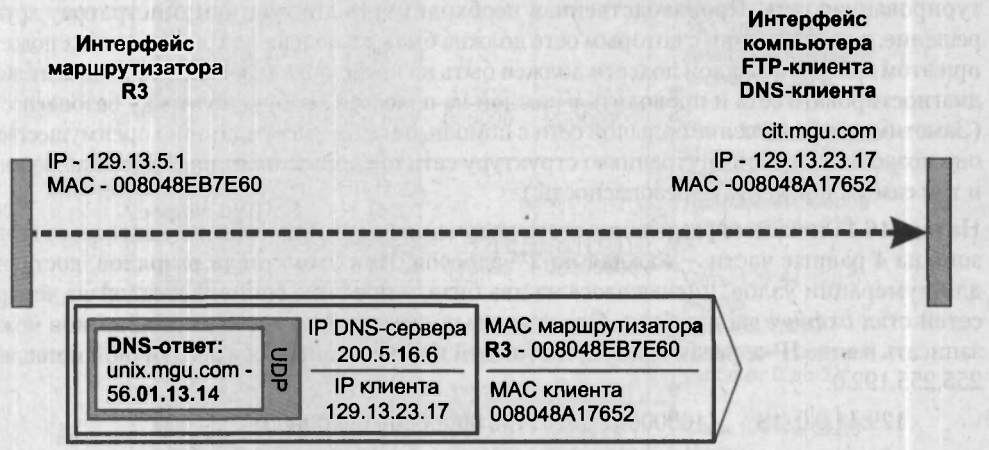


Рис. 16.10. Кадр Ethernet с DNS-ответом, отправленный с маршрутизатора R3 компьютеру-клиенту

FTP-клиент, получив IP-адрес FTP-сервера, посылает ему свое сообщение, используя те же описанные ранее механизмы доставки данных через составную сеть. Однако для читателя будет весьма полезно мысленно воспроизвести этот процесс, обращая особое внимание на значения адресных полей заголовков кадров и заголовка вложенного IP-пакета.

Маршрутизация с использованием масок

Алгоритм маршрутизации усложняется, когда в систему адресации узлов вносятся дополнительные элементы — маски. В чем же причины отказа от хорошо себя зарекомендовавшего в течение многих лет метода адресации, основанного на классах? Основная из них — потребность в структуризации сетей в условиях дефицита нераспределенных номеров сетей.

Часто администраторы сетей испытывают неудобства, поскольку количества централизованно выделенных им номеров сетей недостаточно для того, чтобы структурировать сеть надлежащим образом, например развести все слабо взаимодействующие компьютеры по разным сетям. В такой ситуации возможны два пути. Первый из них связан с получением от какого-либо центрального органа дополнительных номеров сетей. Второй способ, употребляющийся чаще, связан с использованием технологии масок, которая позволяет разделить одну сеть на несколько.

Структуризация сети масками одинаковой длины

Допустим, администратор получил в свое распоряжение сеть класса B: 129.44.0.0. Он может организовать сеть с большим числом узлов, номера которых доступны ему из диапазона 0.0.0.1–0.0.255.254. Всего в его распоряжении имеется (2^{16} – 2) адреса. Вычитание двойки связано с учетом того, что адреса из одних нулей и одних единиц имеют специальное назначение и не годятся для адресации узлов. Однако ему не нужна одна большая неструктурированная сеть. Производственная необходимость диктует администратору другое решение, в соответствии с которым сеть должна быть разделена на три отдельных подсети, при этом трафик в каждой подсети должен быть надежно локализован. Это позволит легче диагностировать сеть и проводить в каждой из подсетей особую политику безопасности. (Заметим, что разделение большой сети с помощью масок имеет еще одно преимущество – оно позволяет скрыть внутреннюю структуру сети предприятия от внешнего наблюдения и тем самым повысить ее безопасность.)

На рис. 16.11 показано разделение всего полученного администратором адресного диапазона на 4 равные части – каждая по 2^{14} адресов. При этом число разрядов, доступное для нумерации узлов, *уменьшилось* на два бита, а префикс (номер) каждой из четырех сетей стал *длиннее* на два бита. Следовательно, каждый из четырех диапазонов можно записать в виде IP-адреса с маской, состоящей из 18 единиц, или в десятичной нотации – 255.255.192.0.

129.44.0.0/18 (10000001 00101100 **00**000000 00000000)

129.44.64.0/18 (10000001 00101100 **01**000000 00000000)

129.44.128.0/18 (10000001 00101100 **10**000000 00000000)

129.44.192.0/18 (10000001 00101100 **11**000000 00000000)

Из приведенных записей видно, что администратор получает возможность использовать для нумерации подсетей два дополнительных бита (выделенных жирным шрифтом). Именно это позволяет ему сделать из одной централизованно выделенной сети четыре, в данном примере это 129.44.0.0/18, 129.44.64.0/18, 129.44.128.0/18, 129.44.192.0/18.

ПРИМЕЧАНИЕ

Некоторые программные и аппаратные маршрутизаторы, следуя устаревшим рекомендациям RFC 950, не поддерживают номера подсетей, которые состоят либо только из одних нулей, либо только из одних единиц. Например, для такого типа оборудования номер сети 129.44.0.0 с маской 255.255.192.0, использованной в нашем примере, окажется недопустимым, поскольку в этом случае разряды в поле номера подсети имеют значение 00. По аналогичным соображениям недопустимым может оказаться номер сети 129.44.192.0 с тем же значением маски. Здесь номер подсети состоит только из единиц. Однако современные маршрутизаторы, поддерживающие рекомендации RFC 1878, свободны от этих ограничений.

Пример сети, построенной путем деления на 4 сети равного размера, показан на рис. 16.12. Весь трафик во внутреннюю сеть 129.44.0.0, направляемый из внешней сети, поступает через маршрутизатор R1. В целях структуризации информационных потоков во внутренней сети установлен дополнительный маршрутизатор R2. Каждая из вновь образованных сетей 129.44.0.0/18, 129.44.64.0/18, 129.44.128.0/18 и 129.44.192.0/18 подключена к соответственно сконфигурированным портам внутреннего маршрутизатора R2.

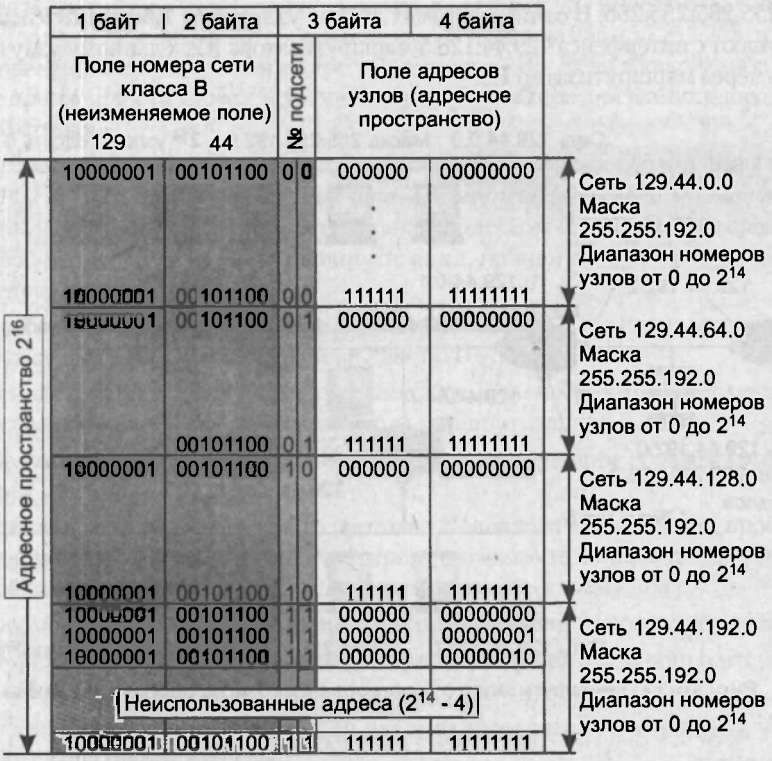


Рис. 16.11. Разделение адресного пространства 129.44.0.0 сети класса B на четыре равные части

ПРИМЕЧАНИЕ

В одной из этих сетей (129.44.192.0/18), выделенной для организации соединения между внешним и внутренним маршрутизаторами, для адресации узлов задействованы всего два адреса — 129.44.192.1 (порт маршрутизатора R2) и 129.44.192.2 (порт маршрутизатора R1). Огромное число узлов в этой подсети не используется. Такой пример выбран исключительно в учебных целях, чтобы показать неэффективность сетей равного размера.

Извне сеть по-прежнему выглядит, как единая сеть класса B. Однако поступающий в сеть общий трафик разделяется локальным маршрутизатором R2 между четырьмя сетями. В условиях, когда механизм классов не действует, маршрутизатор должен иметь другое средство, которое позволило бы ему определять, какая часть 32-разрядного числа, помещенного в поле адреса назначения, является номером сети. Именно этой цели служит дополнительное поле маски, включенное в таблицу маршрутизации (табл. 16.8).

Первые четыре записи в таблице соответствуют внутренним подсетям, непосредственно подключенным к портам маршрутизатора R2.

Запись 0.0.0.0 с маской 0.0.0.0 соответствует маршруту по умолчанию.

Последняя запись определяет специфический маршрут к узлу 129.44.128.15. В тех строках таблицы, в которых в качестве адреса назначения указан полный IP-адрес узла, маска имеет

значение 255.255.255.255. В отличие от всех других узлов сети 129.44.128.0, к которым пакеты поступают с интерфейса 129.44.128.5 маршрутизатора R2, к данному узлу они должны приходить через маршрутизатор R3.

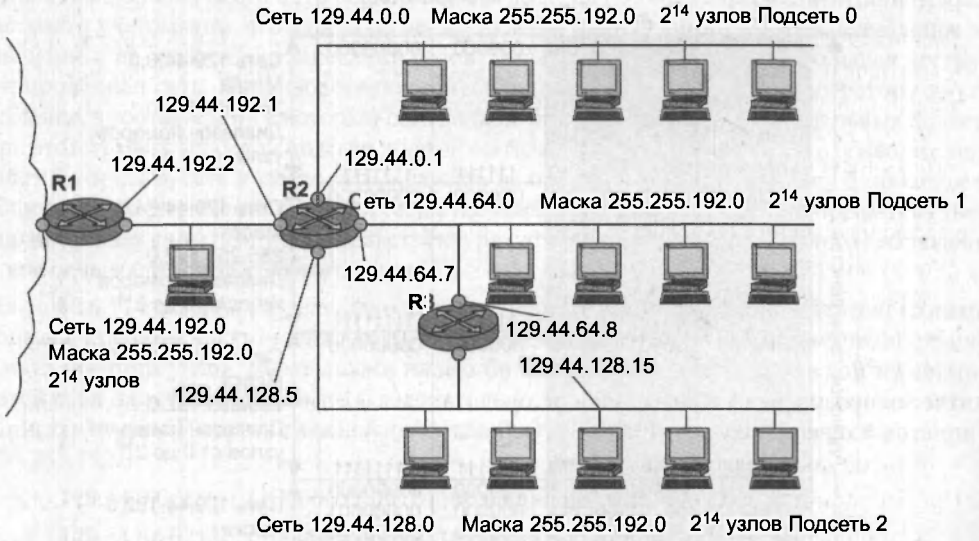


Рис. 16.12. Маршрутизация с использованием масок одинаковой длины

Таблица 16.8. Таблица маршрутизатора R2 в сети с масками одинаковой длины

Адрес назначения	Маска	Адрес следующего маршрутизатора	Адрес порта	Расстояние
129.44.0.0	255.255.192.0	129.44.0.1	129.44.0.1	Подключена
129.44.64.0	255.255.192.0	129.44.64.7	129.44.64.7	Подключена
129.44.128.0	255.255.192.0	129.44.128.5	129.44.128.5	Подключена
129.44.192.0	255.255.192.0	129.44.192.1	129.44.192.1	Подключена
0.0.0.0	0.0.0.0	129.44.192.2	129.44.192.1	–
129.44.128.15	255.255.255.255	129.44.64.8	129.44.64.7	–

Просмотр таблиц маршрутизации с учетом масок

Алгоритм просмотра таблиц маршрутизации, содержащих маски, имеет много общего с описанным алгоритмом просмотра таблиц, не содержащих маски. Однако в нем имеются и существенные изменения.

1. Поиск следующего маршрутизатора для вновь поступившего IP-пакета протокол начинает с того, что *извлекает из пакета адрес назначения* (обозначим его IP_D). Затем протокол IP приступает к процедуре просмотра таблицы маршрутизации, также состоящей из двух фаз, как и процедура просмотра таблицы, в которой столбец маски отсутствует.

2. *Первая фаза* состоит в *поиске специфического маршрута* для адреса IP_D. С этой целью из каждой записи таблицы, в которой маска имеет значение 255.255.255.255, извлекается адрес назначения и сравнивается с адресом из пакета IP_D. Если в какой-либо строке совпадение произошло, то адрес следующего маршрутизатора для данного пакета берется из данной строки.
3. *Вторая фаза* выполняется только в том случае, если во время первой фазы не произошло совпадения адресов. Она состоит в *поиске неспецифического маршрута*, общего для группы узлов, к которой относится и пакет с адресом IP_D. Для этого средствами IP заново просматривается таблица маршрутизации, причем с *каждой* записью производятся следующие действия:
   1) маска (обозначим ее M), содержащаяся в данной записи, «накладывается» на IP-адрес узла назначения IP_D, извлеченный из пакета: IP_D AND M;
   2) полученное в результате число сравнивается со значением, которое помещено в поле адреса назначения той же записи таблицы маршрутизации;
   3) если происходит совпадение, протокол IP соответствующим образом *отмечает эту строку*;
   4) если просмотрены не все строки, то протокол IP аналогичным образом просматривает следующую строку, если все (включая строку о маршруте по умолчанию), то просмотр записей заканчивается, и происходит переход к следующему шагу.
4. После просмотра всей таблицы маршрутизатор выполняет одно их трех действий:
   1) если не произошло ни одного совпадения и маршрут по умолчанию отсутствует, то пакет отбрасывается;
   2) если произошло одно совпадение, то пакет отправляется по маршруту, указанному в строке с совпавшим адресом;
   3) если произошло несколько совпадений, то все помеченные строки сравниваются и выбирается маршрут из той строки, в которой количество совпавших двоичных разрядов наибольшее (другими словами, в ситуации, когда адрес назначения пакета принадлежит сразу нескольким подсетям, маршрутизатор использует наиболее специфический маршрут).

ПРИМЕЧАНИЕ

Во многих таблицах маршрутизации запись с адресом 0.0.0.0 и маской 0.0.0.0 соответствует маршруту по умолчанию. Действительно, любой адрес в пришедшем пакете после наложения на него маски 0.0.0.0 даст адрес сети 0.0.0.0, что совпадает с адресом, указанным в записи. Поскольку маска 0.0.0.0 имеет нулевую длину, то этот маршрут считается самым неспецифическим и используется только при отсутствии совпадений с остальными записями из таблицы маршрутизации.

Проиллюстрируем, как маршрутизатор R2 (см. рис. 16.12) использует описанный алгоритм для работы со своей таблицей маршрутизации (см. табл. 16.8). Пусть на маршрутизатор R2 поступает пакет с адресом назначения 129.44.78.200. Модуль IP, установленный на этом маршрутизаторе, прежде всего сравнит этот адрес с адресом 129.44.128.15, для которого определен специфический маршрут. Совпадения нет, поэтому модуль IP начинает последовательно обрабатывать все строки таблицы, накладывая маски и сравнивая результаты до тех пор, пока не найдет совпадения номера сети в адресе назначения и в строке таблицы. В результате определяется маршрут для пакета 129.44.78.200 — он должен быть отправлен на выходной порт маршрутизатора 129.44.64.7 в сеть 129.44.64.0, непосредственно подключенную к данному маршрутизатору.

Использование масок переменной длины

Во многих случаях более эффективным является разбиение сети на подсети разного размера. В частности, для подсети, которая связывает два маршрутизатора по двухточечной схеме, даже количество адресов сети класса C явно является избыточным.

На рис. 16.13 приведен другой пример распределения того же адресного пространства 129.44.0.0/16, что и в предыдущем примере. Здесь половина из имеющихся адресов (2^{15}) отведена для создания *сети 1*, имеющей адрес 129.44.0.0 и маску 255.255.128.0.

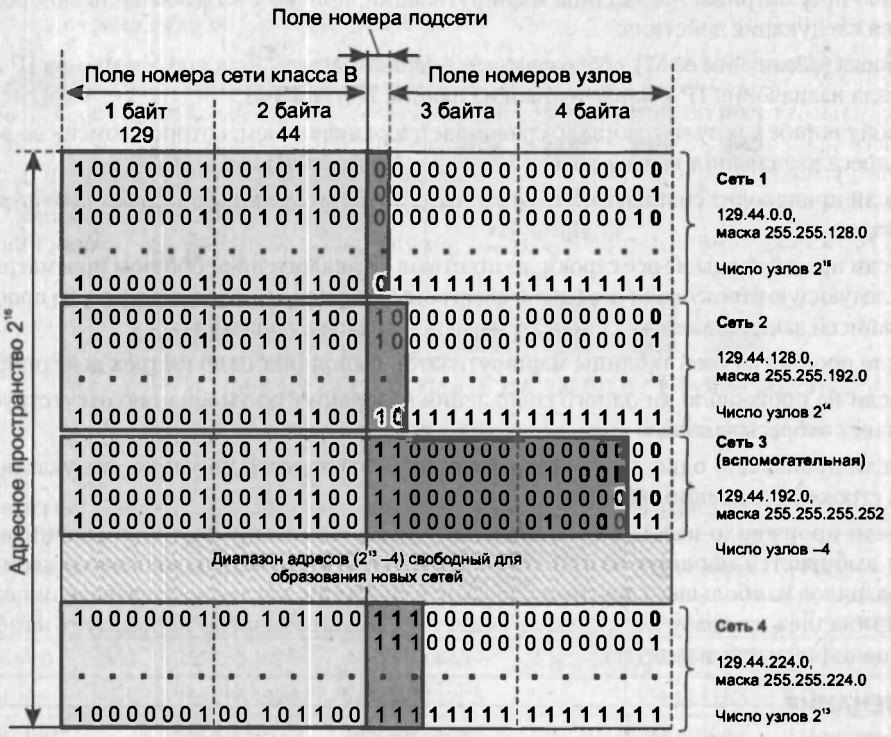


Рис. 16.13. Разделение адресного пространства 129.44.0.0 сети класса B на сети разного размера путем использования масок переменной длины

Следующая порция адресов, составляющая четверть всего адресного пространства (2^{14}), назначена для *сети 2* 129.44.128.0 с маской 255.255.192.0.

Далее в пространстве адресов был «вырезан» небольшой фрагмент для создания вспомогательной *сети 3*, предназначенной для связывания внутреннего маршрутизатора R2 с внешним маршрутизатором R1. Для нумерации узлов в такой вырожденной сети достаточно отвести два двоичных разряда. Из четырех возможных комбинаций номеров узлов: 00, 01, 10 и 11 два номера имеют специальное назначение и не могут быть присвоены узлам, но оставшиеся два 10 и 01 позволяют адресовать порты маршрутизаторов. Поле номера узла в таком случае имеет два двоичных разряда, маска в десятичной нотации имеет вид 255.255.255.252, а номер сети, как видно из рисунка, равен 129.44.192.0.

ПРИМЕЧАНИЕ

Глобальным связям между маршрутизаторами, соединенными по двухточечной схеме, не обязательно давать IP-адреса. Однако чаще всего такой вырожденной сети все же дают IP-адрес. Помимо прочего, это делается, например, для того, чтобы скрыть внутреннюю структуру сети и обращаться к ней по одному адресу входного порта маршрутизатора, в данном примере по адресу 129.44.192.1, применяя технику трансляции сетевых адресов (Network Address Translation, NAT[1]).

Оставшееся адресное пространство администратор может «нарезать» на разное количество сетей разного объема в зависимости от своих потребностей. Из оставшегося пула ($2^{14} - 4$) адресов администратор, например, может образовать еще одну достаточно большую сеть с числом узлов 2^{13} — на рисунке это *сеть 4*. При этом свободными останутся почти столько же адресов ($2^{13} - 4$), которые также могут быть использованы для создания новых сетей. К примеру, из этого «остатка» можно образовать 31 сеть, каждая из которых равна размеру сети класса C, и к тому же еще несколько сетей меньшего размера. Ясно, что разбиение может быть другим, но в любом случае с помощью масок переменного размера администратор имеет больше возможностей рационально использовать все имеющиеся у него адреса.

На рис. 16.14 показан пример сети, структурированной с помощью масок переменной длины.

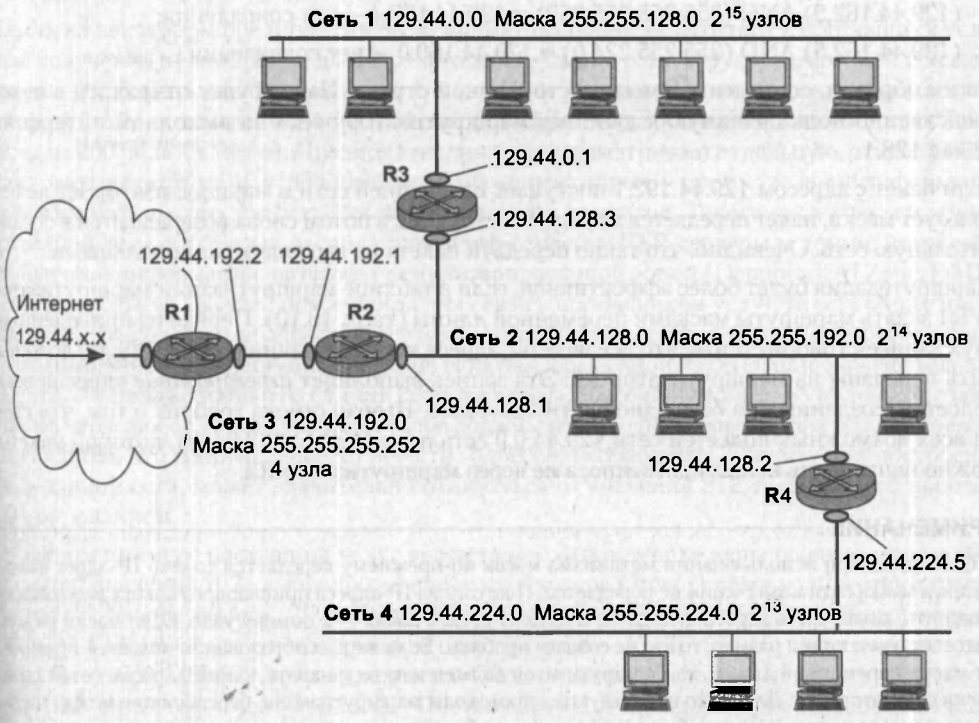


Рис. 16.14. Структуризация сети масками переменной длины

[1] О технологии NAT читайте в главе 18.

Давайте посмотрим, как маршрутизатор R2 обрабатывает поступающие на его интерфейсы пакеты (табл. 16.9).

Таблица 16.9. Таблица маршрутизатора R2 в сети с масками переменной длины

Адрес назначения	Маска	Адрес следующего маршрутизатора	Адрес порта	Расстояние
129.44.0.0	255.255.128.0	129.44.128.3	129.44.128.1	1
129.44.128.0	255.255.192.0	129.44.128.1	129.44.128.1	Подключена
129.44.192.0	255.255.255.252	129.44.192.1	129.44.192.1	Подключена
129.44.224.0	255.255.224.0	129.44.128.2	129.44.128.1	1
0.0.0.0	0.0.0.0	129.44.192.2	129.44.192.1	–

Пусть поступивший на R2 пакет имеет адрес назначения 129.44.162.5. Поскольку специфические маршруты в таблице отсутствуют, маршрутизатор переходит ко второй фазе — фазе последовательного анализа строк на предмет поиска совпадения с адресом назначения:

- (129.44.162.5) AND (255.255.128.0) = 129.44.128.0 — нет совпадения;
- (129.44.162.5) AND (255.255.192.0) = 129.44.128.0 — совпадение;
- (129.44.162.5) AND (255.255.255.252) = 129.44.162.4 — нет совпадения;
- (129.44.162.5) AND (255.255.224.0) = 129.44.160.0 — нет совпадения.

Таким образом, совпадение имеет место в одной строке. Пакет будет отправлен в непосредственно подключенную к данному маршрутизатору сеть на выходной интерфейс 129.44.128.1.

Если пакет с адресом 129.44.192.1 поступает из внешней сети и маршрутизатор R1 не использует маски, пакет передается маршрутизатору R2, а потом снова возвращается в соединительную сеть. Очевидно, что такие передачи пакета не выглядят рациональными.

Маршрутизация будет более эффективной, если в таблице маршрутизации маршрутизатора R1 задать маршруты масками переменной длины (табл. 16.10). Первая из приведенных двух записей говорит о том, что все пакеты, адреса которых начинаются с 129.44, должны быть переданы на маршрутизатор R2. Эта запись выполняет *агрегирование* адресов всех подсетей, созданных на базе одной сети 129.44.0.0. Вторая строка говорит о том, что среди всех возможных подсетей сети 129.44.0.0 есть одна (129.44.192.0/30), которой пакеты можно направлять непосредственно, а не через маршрутизатор R2.

ПРИМЕЧАНИЕ

В IP-пакетах при использовании механизма масок по-прежнему передается только IP-адрес назначения, а маска сети назначения не передается. Поэтому из IP-адреса пришедшего пакета невозможно выяснить, какая часть адреса относится к номеру сети, а какая — к номеру узла. Если маски во всех подсетях имеют один размер, то это не создает проблем. Если же для образования подсетей применяют маски переменной длины, то маршрутизатор должен как-то узнавать, каким адресам сетей какие маски соответствуют. Для этого используются протоколы маршрутизации, переносящие между маршрутизаторами не только служебную информацию об адресах сетей, но и о масках, соответствующих этим номерам. К таким протоколам относятся протоколы RIPv2 и OSPF, а вот, например, протокол RIP маски не переносит и для маршрутизации на основе масок переменной длины не подходит.

Таблица 16.10. Фрагмент таблицы маршрутизатора R1

Адрес назначения	Маска	Адрес следующего маршрутизатора	Адрес порта	Расстояние
129.44.0.0	255.255.0.0	129.44.192.1	129.44.191.2	2
129.44.192.0	255.255.255.192	129.44.192.2	129.44.192.2	Подключена

Перекрытие адресных пространств

Со сложностями использования масок администратор впервые сталкивается не тогда, когда начинает конфигурировать сетевые интерфейсы и создавать таблицы маршрутизации, а гораздо раньше — на этапе планирования сети. Планирование включает определение количества сетей, из которых будет состоять корпоративная сеть, оценку требуемого количества адресов для каждой сети, получение пула адресов от поставщика услуг, распределение адресного пространства между сетями. Последняя задача часто оказывается нетривиальной, особенно когда решается в условиях дефицита адресов.

Рассмотрим пример использования масок для организации *перекрывающихся адресных пространств*.

Пусть на некотором предприятии было принято решение обратиться к поставщику услуг для получения пула адресов, достаточного для создания сети, структура, которой показана на рис. 16.15. Сеть клиента включает три подсети. Две из них — это надежно защищенные от внешних атак внутренние сети отделов: сеть Ethernet на 600 пользователей и сеть Token Ring на 200 пользователей. Предприятие также предусматривает отдельную, открытую для доступа извне сеть на 10 узлов, главное назначение которой — предоставление информации в режиме открытого доступа для потенциальных клиентов. Такого рода участки корпоративной сети, в которых располагаются веб-серверы, FTP-серверы и другие источники публичной информации, называют **демилитаризованной зоной** (Demilitarized Zone, DMZ). Еще одна сеть на два узла потребуется для связи с поставщиком услуг, то есть общее число адресов, требуемых для адресации сетевых интерфейсов, составляет 812. Кроме того, необходимо, чтобы пул доступных адресов включал для каждой из сетей широковещательные адреса, состоящие только из единиц, а также адреса, состоящие только из нулей. Учитывая также, что в любой сети адреса всех узлов должны иметь одинаковые префиксы, становится очевидным, что минимальное количество адресов, необходимое клиенту для построения задуманной сети, может значительно отличаться от значения 812, полученного простым суммированием.

В данном примере поставщик услуг решает выделить клиенту непрерывный пул из 1024 адресов. Значение 1024 выбрано как наиболее близкое к требуемому количеству адресов, равному степени двойки ($2^{10} = 1024$). Поставщик услуг выполняет поиск области такого размера в имеющемся у него адресном пространстве — 131.57.0.0/16, часть которого, как показано на рис. 16.16, уже распределена. Обозначим распределенные участки и владеющих ими клиентов через S1, S2 и S3. Поставщик услуг находит среди нераспределенных еще адресов непрерывный участок размером 1024 адреса, начальный адрес которого кратен размеру данного участка. Таким образом, наш клиент получает пул адресов 131.57.8.0/22, обозначенный на рисунке через S.

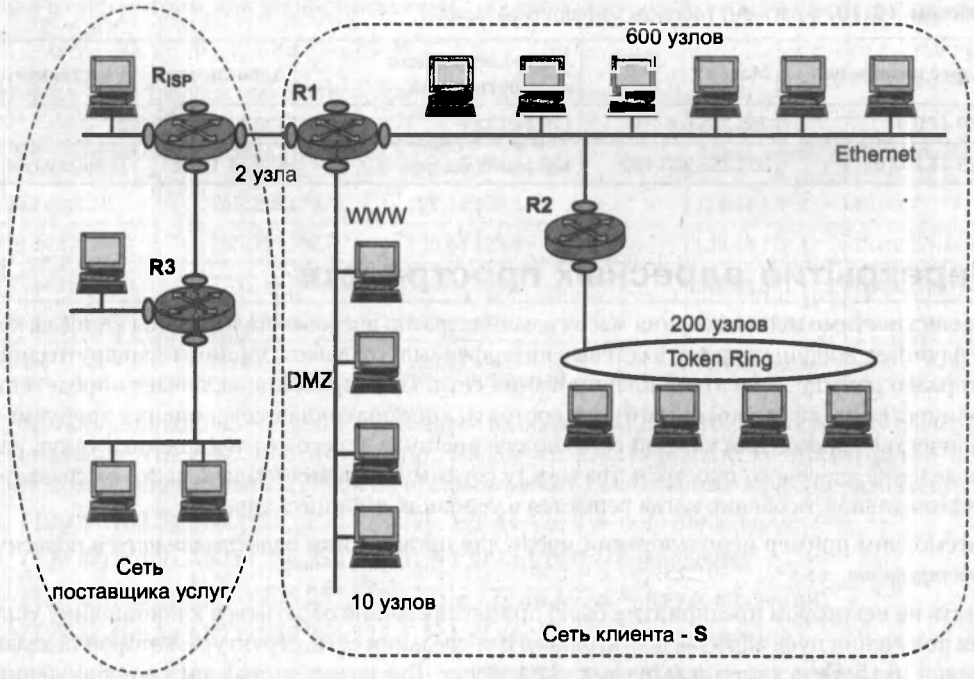


Рис. 16.15. Сети поставщика услуг и клиента

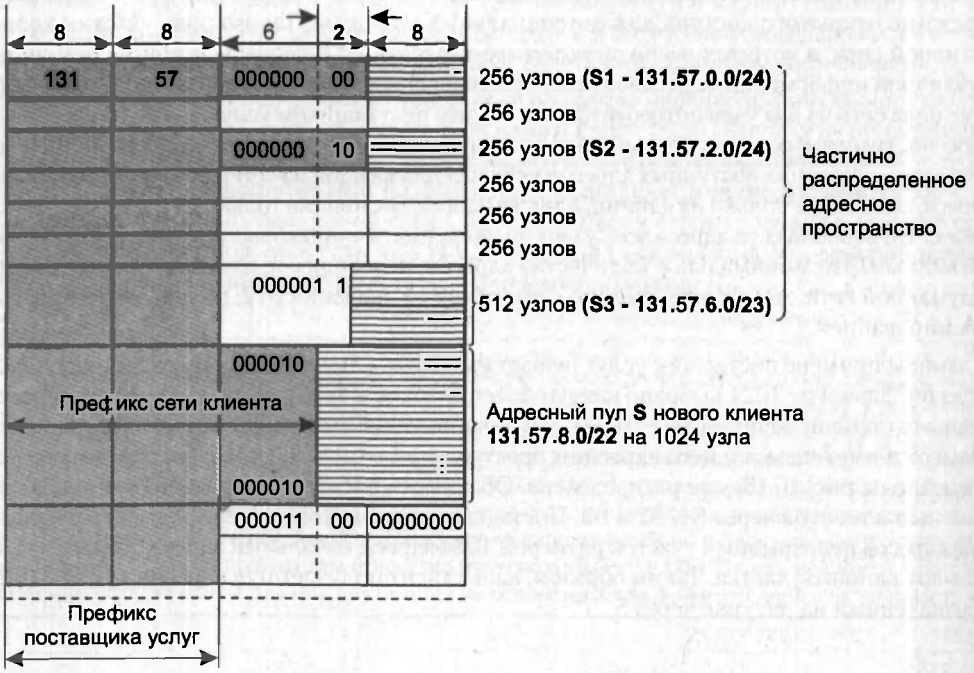


Рис. 16.16. Адресное пространство поставщика услуг

Далее начинается самый сложный этап — распределение полученного от поставщика услуг адресного пула S между четырьмя сетями клиента. Прежде всего, администратор решил назначить для самой большой сети (Ethernet на 600 узлов) весь пул адресов 131.57.8.0/22, полученный от поставщика услуг (рис. 16.17). Номер, назначенный для этой сети, совпадает с номером сети, полученным от поставщика услуг. А как же быть с оставшимися тремя сетями? Администратор учел, что для сети Ethernet требуется только 600 адресов, а из оставшихся 624 «выкроил» сеть Token Ring 131.57.9.0/24 на 250 адресов. Воспользовавшись тем, что для Token Ring требуется только 200 адресов, он «вырезал» из нее два участка: для сети DMZ 131.57.9.16/28 на 16 адресов и для связывающей сети 131.57.9.32/30 на 4 адреса. В результате все сети клиента получили достаточное (а иногда и с избытком) количество адресов.

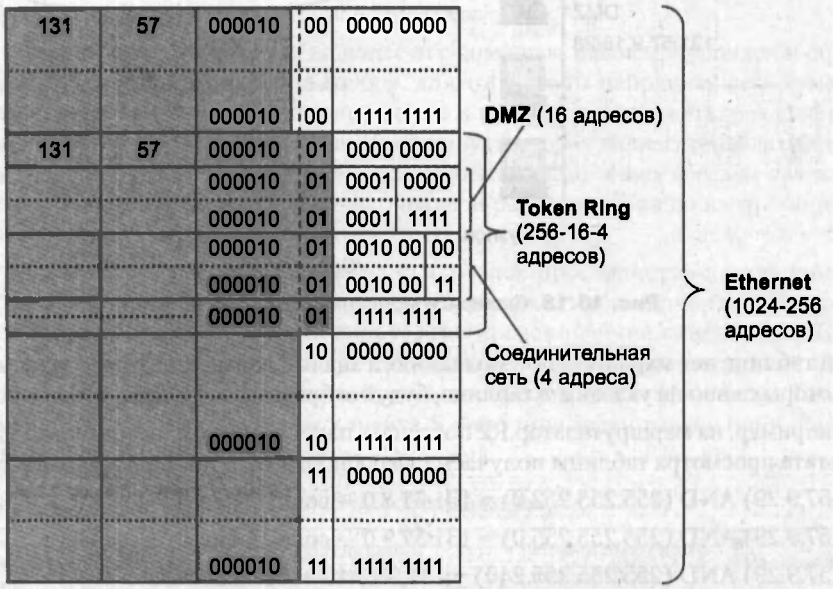


Рис. 16.17. Планирование адресного пространства для сетей клиента

Следующий этап — это конфигурирование сетевых интерфейсов конечных узлов и маршрутизаторов. Каждому интерфейсу сообщается его IP-адрес и соответствующая маска. На рис. 16.18. показана сконфигурированная сеть клиента.

После конфигурирования сетевых интерфейсов должны быть созданы таблицы маршрутизации маршрутизаторов R1 и R2 клиента. Они могут быть сгенерированы автоматически или с участием администратора. Таблица маршрутизации маршрутизатора R2 соответствует табл. 16.11.

Таблица 16.11. Таблица маршрутизации маршрутизатора R2

Адрес назначения	Маска	Адрес следующего маршрутизатора	Адрес выходного интерфейса	Расстояние
131.57.8.0	255.255.252.0	131.57.8.2	131.57.8.2	Подключена
131.57.9.0	255.255.255.0	131.57.9.1	131.57.9.1	Подключена
131.57.9.16	255.255.255.240	131.57.8.1	131.57.8.2	1
131.57.9.32	255.255.255.252	131.57.8.1	131.57.8.2	1

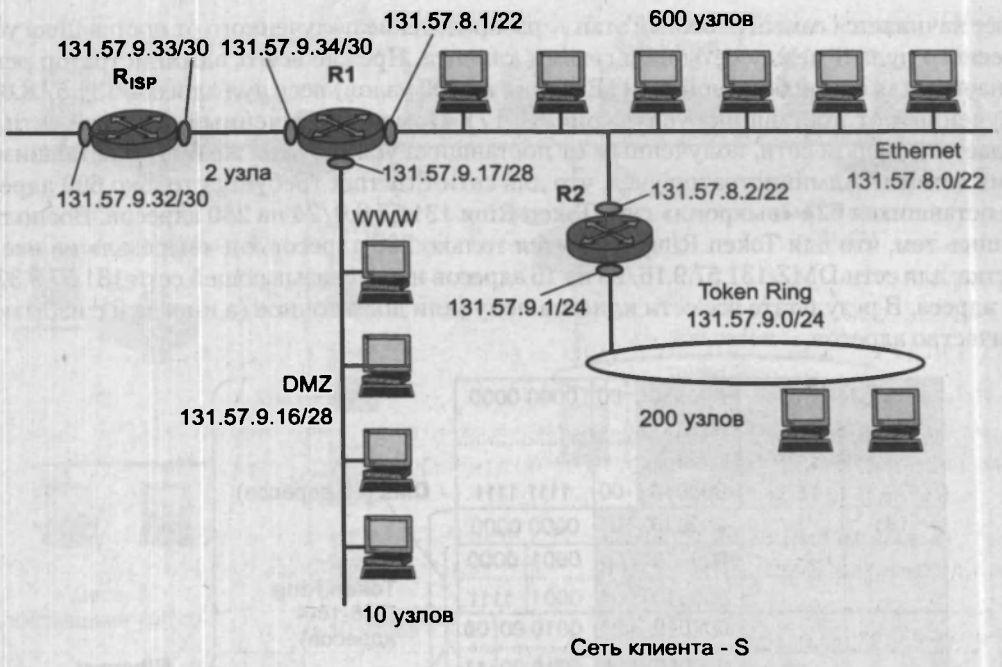


Рис. 16.18. Сконфигурированная сеть клиента

В данной таблице нет маршрута по умолчанию, а значит, все пакеты, адресованные сетям, адреса которых явно не указаны в таблице, будут отбрасываться маршрутизатором.

Пусть, например, на маршрутизатор R2 поступает пакет с адресом назначения 131.57.9.29. В результате просмотра таблицы получаем следующие результаты для каждой строки:

- (131.57.9.29) AND (255.255.252.0) = 131.57.8.0 — совпадение;
- (131.57.9.29) AND (255.255.255.0) = 131.57.9.0 — совпадение;
- (131.57.9.29) AND (255.255.255.240) = 131.57.9.16 — совпадение;
- (131.57.9.29) AND (255.255.255.252) = 131.57.9.28 — нет совпадения.

Поскольку при наличии нескольких совпадений выбирается маршрут из той строки, в которой совпадение адреса назначения с адресом из пакета имеет наибольшую длину, определено, что пакет с адресом 131.57.9.29 направляется в сеть DMZ.

CIDR

За последние несколько лет в Интернете многое изменилось: резко возросло число узлов и сетей, повысилась интенсивность трафика, изменился характер передаваемых данных. Из-за несовершенства протоколов маршрутизации обмен сообщениями об обновлении таблиц стал приводить к сбоям магистральных маршрутизаторов, происходящим из-за перегрузок при обработке большого объема служебной информации. Так, сегодня таблицы магистральных маршрутизаторов в Интернете могут содержать до нескольких сотен и даже тысяч маршрутов.

На решение этой проблемы направлена, в частности, и технология **бесклассовой междоменной маршрутизации** (Classless Inter-Domain Routing, CIDR).

Суть технологии CIDR заключается в следующем. Каждому поставщику услуг Интернета назначается *непрерывный* диапазон IP-адресов. При таком подходе все адреса каждого поставщика услуг имеют общую старшую часть — **префикс**, поэтому маршрутизация на магистралях Интернета может осуществляться на основе префиксов, а не полных адресов сетей. А это значит, что вместо множества записей по числу сетей будет достаточно поместить *одну запись сразу для всех сетей, имеющих общий префикс*. Такое агрегирование адресов позволит уменьшить объем таблиц в маршрутизаторах всех уровней, а следовательно, ускорить работу маршрутизаторов и повысить пропускную способность Интернета.

Ранее мы рассматривали примеры, где администраторы корпоративных сетей с помощью масок делили на несколько частей непрерывный пул адресов, полученный от поставщика услуг, чтобы использовать эти части для структуризации своей сети. Такой вариант применения масок называется *разделением на подсети*.

Вместе с тем в процессе разделения на подсети с помощью масок проявлялся и обратный эффект их применения. Упрощенно говоря, для того чтобы направить весь суммарный трафик, адресованный из внешнего окружения в корпоративную сеть, разделенную на подсети, достаточно, чтобы во всех внешних маршрутизаторах наличествовала одна строка. В этой строке на месте адреса назначения должен быть указан *общий префикс для всех этих сетей*. Здесь мы имеем дело с операцией, обратной разделению на подсети — операцией *агрегирования несколько сетей в одну более крупную*.

Вернемся к рис. 16.16, на котором показано адресное пространство поставщика услуг с участками S1, S2, S3 и S, переданными в пользование четырем клиентам. Этот пример также иллюстрирует рис. 16.19. В результате агрегирования сетей клиентов в табл. 16.12 маршрутизатора R_{ISP} поставщика услуг для каждого клиента будет выделено по одной строке независимо от количества подсетей, организованных ими в своих сетях. Так, вместо четырех маршрутов к четырем сетям клиента S в таблице задан только один общий для всех них маршрут (выделенный жирным шрифтом).

Таблица 16.12. Таблица маршрутизатора R_{ISP} поставщика услуг

Адрес назначения	Маска	Следующий маршрутизатор	Номер выходного интерфейса	Расстояние
131.57 0.0 (S1)	255.255.255.0	R3	1	Подключена
131.57 2.0 (S2)	255.255.255.0	R3	3	1
131.57 4.0 (S3)	255.255.254.0	R1	3	1
131.57.8.0 (S)	**255.255.252.0**	**R1**	**2**	**Подключена**
Маршрут по умолчанию	0.0.0.0	$R_{external}$	4	—

Итак, внедрение технологии CIDR позволяет решить две основные задачи.

- *Более экономное расходование адресного пространства.* Благодаря технологии CIDR поставщики услуг получают возможность «нарезать» блоки из выделенного им адресного пространства в точном соответствии с требованиями каждого клиента, при этом у клиента остается пространство для маневра на случай будущего роста.
- *Уменьшение числа записей в таблицах маршрутизации* за счет объединения маршрутов — одна запись в таблице маршрутизации может представлять большое количество сетей. Если все поставщики услуг Интернета начнут придерживаться стратегии CIDR, то особенно заметный выигрыш будет достигаться в магистральных маршрутизаторах.

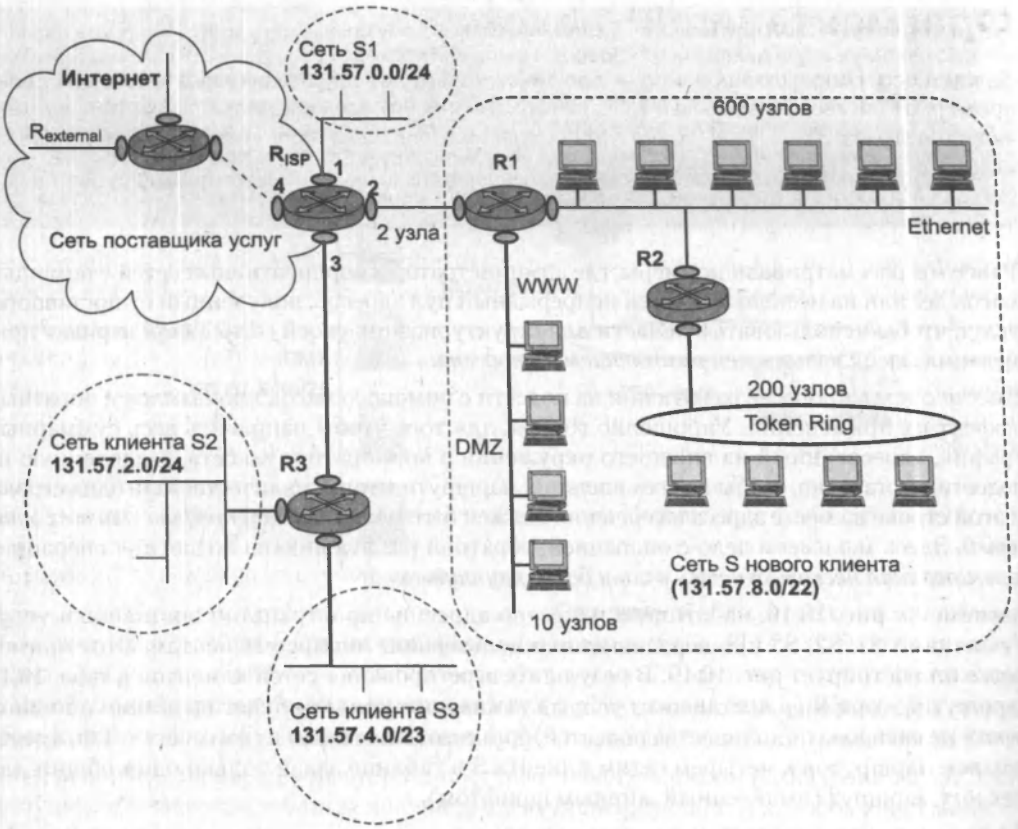


Рис. 16.19. Объединение подсетей

Необходимым условием эффективного использования технологии CIDR является **локализация адресов**, то есть назначение адресов, имеющих совпадающие префиксы, сетям, располагающимся территориально по соседству. Только в таком случае трафик может быть агрегирован.

К сожалению, сейчас распределение адресов носит во многом случайный характер. Кардинальный путь решения проблемы — перенумерование сетей. Однако эта процедура сопряжена с определенными временными и материальными затратами, и для ее проведения пользователей нужно каким-либо образом стимулировать. В качестве таких стимулов рассматривается, например, введение оплаты за строку в таблице маршрутизации или же за количество узлов в сети. Первое требование подводит потребителя к мысли получить у поставщика услуг такой адрес, чтобы маршрутизация трафика в его сеть шла на основании префикса, и номер его сети не фигурировал больше в магистральных маршрутизаторах. Требование оплаты каждого адреса узла также может подтолкнуть пользователя решиться на перенумерование с тем, чтобы получить ровно столько адресов, сколько ему нужно.

Технология CIDR уже успешно используется в текущей версии протокола IP (IPv4) и поддерживается такими протоколами маршрутизации, как OSPF, RIP-2, BGP4 (в основном на магистральных маршрутизаторах Интернета). Особенности применения технологии CIDR в новой версии протокола IP (IPv6) будут рассмотрены в главе 18.

Фрагментация IP-пакетов

Важной особенностью протокола IP, отличающей его от других сетевых протоколов (например, от сетевого протокола IPX, который какое-то время назад конкурировал с IP), является его способность выполнять *динамическую фрагментацию* пакетов при передаче их между сетями с различными максимально допустимыми значениями длины поля данных кадров (Maximum Transmission Unit, **MTU**). Значения MTU зависят как от протокола, так и от настройки сетевых интерфейсов.

Прежде всего отметим разницу между фрагментацией сообщений *в узле-отправителе* и динамической фрагментацией сообщений *в транзитных узлах* сети — маршрутизаторах.

В первом случае деление сообщения на несколько более мелких частей (фрагментация) происходит при передаче данных между протоколами одного и того же стека внутри компьютера. Протоколы, выполняющие фрагментацию в пределах узла, анализируют тип технологии нижнего уровня, определяют ее MTU и делят сообщения на такие части, которые умещаются в кадры канального уровня того же стека протоколов.

В стеке TCP/IP эту задачу решает протокол TCP, который разбивает поток байтов, передаваемый ему с прикладного уровня, на сегменты нужного размера, например, по 1460 байт, если на нижнем уровне данной сети работает протокол Ethernet. Протокол IP в узле-отправителе, как правило, не использует свои возможности по фрагментации пакетов.

А вот на транзитном узле — маршрутизаторе, когда пакет необходимо передать из сети с большим значением MTU в сеть с меньшим значением MTU, способности протокола IP выполнять фрагментацию становятся востребованными. *Пакеты-фрагменты*, путешествуя по сети, могут вторично подвергнуться фрагментации на каком-либо из промежуточных маршрутизаторов.

Параметры фрагментации

Каждый из фрагментов должен быть снабжен полноценным заголовком IP. Некоторые из полей заголовка (идентификатор, TTL, флаги DF и MF, смещение) непосредственно предназначены для последующей *сборки* фрагментов в исходное сообщение.

- **Идентификатор** пакета используется для *распознавания* пакетов, образовавшихся путем деления на части (фрагментации) исходного пакета. Все части (фрагменты) одного пакета должны иметь одинаковое значение этого поля. Модуль IP, отправляющий пакет, устанавливает в поле идентификатора значение, которое должно быть уникальным для данной пары отправителя и получателя в течение всего времени, пока данный пакет (или любой его фрагмент) может существовать в составной IP-сети.
- Поле **времени жизни** (Time To Live, TTL) занимает один байт и определяет предельный срок, в течение которого пакет может перемещаться по сети. Время жизни пакета измеряется в секундах и задается источником (отправителем). Как уже отмечалось в начале этой главы, по истечении каждой секунды пребывания на каждом из маршрутизаторов, через которые проходит пакет во время своего «путешествия» по сети, из его текущего времени жизни вычитается единица; единица вычитается и в том случае, если время пребывания было меньше секунды. Поскольку современные маршрутизаторы редко обрабатывают пакет дольше, чем за одну секунду, то время жизни можно интерпретировать как максимальное число транзитных узлов, которые разрешено пройти пакету. Если значение поля времени жизни становится нулевым до того, как пакет достигает

получателя, пакет уничтожается. При сборке фрагментов хост-получатель использует значение TTL как крайний срок ожидания недостающих фрагментов.

- Поле **смещения фрагмента** предоставляет получателю информацию о положении фрагмента относительно начала поля данных исходного нефрагментированного пакета. Так, например, первый фрагмент будет иметь в поле смещения нулевое значение. В пакете, не разбитом на фрагменты, поле смещения также имеет нулевое значение. Смещение задается в байтах и должно быть кратно 8 байт.
- Установленный в единицу однобитный флаг **MF** (More Fragments — больше фрагментов) говорит о том, что данный пакет является промежуточным (не последним) фрагментом. Модуль IP, отправляющий нефрагментированный пакет, устанавливает бит MF в нуль.
- Флаг **DF** (Do not Fragment — не фрагментировать), установленный в единицу, запрещает маршрутизатору фрагментировать данный пакет. Если помеченный таким образом пакет не может достигнуть получателя без фрагментации, то модуль IP его уничтожает, а узлу-отправителю посылается диагностическое сообщение.

ПРИМЕЧАНИЕ

Возможность запретить фрагментацию позволяет в некоторых случаях ускорить работу приложений. Для этого вначале необходимо исследовать сеть, определить максимальный размер пакета, который сможет пройти весь путь без фрагментации, а затем использовать пакеты такого или меньшего размера для обмена данными. Данная возможность позволяет также предотвратить фрагментацию в тех случаях, когда хост-получатель не имеет достаточных ресурсов для сборки фрагментов.

Механизм фрагментации

Рассмотрим механизм фрагментации на примере составной сети, показанной на рис. 16.20.

В одной из подсетей (Frame Relay) значение MTU равно 4080, в другой (Ethernet) — 1492. Хост, принадлежащий сети Frame Relay, передает данные хосту в сети Ethernet. На обоих хостах, а также на маршрутизаторе, связывающем эти подсети, установлен стек протоколов TCP/IP.

Транспортному уровню *хоста-отправителя* известно значение MTU нижележащей технологии (4080). На основании этого модуль TCP и «нарезает» свои сегменты размером 4000 байт и передает вниз протоколу IP, который помещает сегменты в поле данных IP-пакетов и генерирует для них заголовки. Обратим особое внимание на заполнение тех полей заголовка, которые прямо связаны с фрагментацией:

- пакету присваивается уникальный *идентификатор*, например 12456;
- поскольку пакет пока еще не был фрагментирован, в поле *смещения* помещается значение 0;
- признак *MF* также обнуляется, это показывает, что пакет одновременно является и своим последним фрагментом;
- признак *DF* устанавливается в 1, это означает, что данный пакет можно фрагментировать.

Общая величина IP-пакета составляет 4000 плюс 20 (размер заголовка IP), то есть 4020 байт, что умещается в поле данных кадра Frame Relay, которое в данном примере равно 4080. Далее модуль IP хоста-отправителя передает этот кадр своему сетевому интерфейсу Frame Relay, который отправляет кадры следующему маршрутизатору.

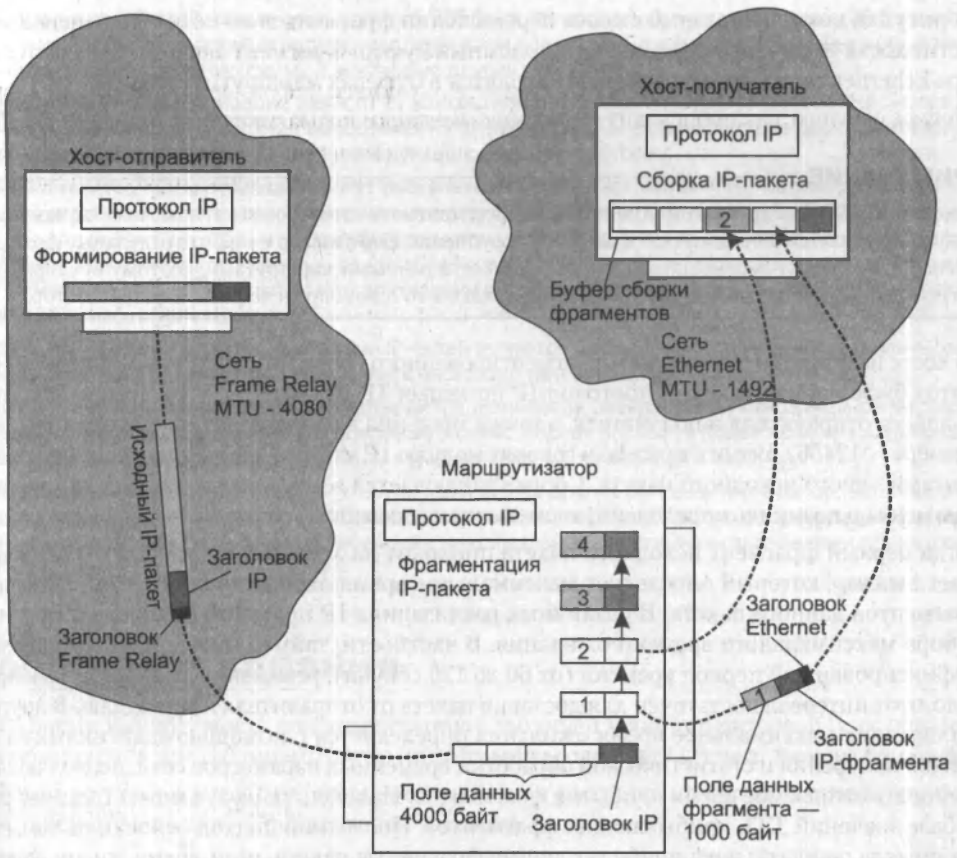


Рис. 16.20. Фрагментация

Модуль IP маршрутизатора по сетевому адресу прибывшего IP-пакета определяет, что пакет нужно передать в сеть Ethernet. Однако она имеет значение MTU, равное 1492, что значительно меньше размера поступившего на входной интерфейс пакета. Следовательно, IP-пакет необходимо фрагментировать. Модуль IP выбирает размер поля данных фрагмента равным 1000, так что из одного большого IP-пакета получается 4 маленьких пакета-фрагмента. Для каждого фрагмента и его заголовка IP в маршрутизаторе создается отдельный буфер (на рисунке фрагменты и соответствующие им буферы пронумерованы от 1 до 4). Протокол IP копирует в эти буферы содержимое некоторых полей заголовка IP исходного пакета, создавая тем самым «заготовки» заголовков IP всех новых пакетов-фрагментов. Одни параметры заголовка IP копируются в заголовки всех фрагментов, другие — лишь в заголовок первого фрагмента.

В процессе фрагментации могут измениться значения некоторых полей заголовков IP в пакетах-фрагментах по сравнению с заголовком IP исходного пакета. Так, каждый фрагмент имеет собственные значения контрольной суммы заголовка, смещения фрагмента и общей длины пакета. Во всех пакетах, кроме последнего, флаг MF устанавливается в единицу, а в последнем фрагменте — в нуль. Полученные пакеты-фрагменты имеют длину 1020 байт (с учетом заголовка IP), поэтому они свободно помещаются в поле данных кадров Ethernet.

На рисунке показаны разные стадии перемещения фрагментов по сети. Фрагмент 2 уже достиг хоста-получателя и помещен в приемный буфер. Фрагмент 1 еще перемещается по сети Ethernet, остальные фрагменты находятся в буферах маршрутизатора.

А теперь обсудим, как происходит *сборка фрагментированного пакета на хосте назначения.*

ПРИМЕЧАНИЕ

Отметим, что IP-маршрутизаторы не собирают фрагменты пакетов в более крупные пакеты, даже если на пути встречается сеть, допускающая такое укрупнение. Это связано с тем, что отдельные фрагменты сообщения могут перемещаться по составной сети разными маршрутами, поэтому нет гарантии, что все фрагменты на своем пути пройдут через какой-то один определенный маршрутизатор.

На хосте назначения для каждого фрагментированного пакета отводится отдельный буфер. В этот буфер принимающий протокол IP помещает IP-фрагменты, у которых совпадают IP-адреса отправителя и получателя, а также значения в полях идентификатора (в нашем примере — 12456). Все эти признаки говорят модулю IP, что данные пакеты являются фрагментами одного исходного пакета. Сборка заключается в помещении данных из каждого фрагмента в позицию, определенную *смещением*, указанным в заголовке фрагмента.

Когда первый фрагмент исходного пакета приходит на хост-получатель, этот хост запускает *таймер*, который определяет максимальное время ожидания прибытия остальных фрагментов данного пакета. В различных реализациях IP применяются разные правила выбора максимального времени ожидания. В частности, таймер может быть установлен на фиксированный период времени (от 60 до 120 секунд), рекомендуемый RFC. Как правило, этот интервал достаточен для доставки пакета от отправителя получателю. В других реализациях максимальное время ожидания определяется с помощью адаптивных алгоритмов измерения и статистической обработки временны́х параметров сети, позволяющих оценивать ожидаемое время прибытия фрагментов. Наконец, тайм-аут может быть выбран на базе значений TTL прибывающих фрагментов. Последний подход основан на том, что нет смысла ожидать, пока прибудут другие фрагменты пакета, если время жизни одного из прибывших фрагментов уже истекло.

ПРИМЕЧАНИЕ

Если хотя бы один фрагмент пакета не успеет прийти на хост назначения к моменту истечения таймера, то никаких действий по дублированию отсутствующего фрагмента не предпринимается, а все полученные к этому времени фрагменты пакета отбрасываются! Хосту, пославшему исходный пакет, направляется ICMP-сообщение об ошибке. Такому поведению протокола IP вполне соответствует его кредо «с максимальными усилиями» — стараться по возможности, но никаких гарантий не давать.

Признаком окончания сборки является отсутствие незаполненных промежутков в поле данных и прибытие последнего фрагмента (с равным нулю флагом MF) до истечения тайм-аута. После того как данные собраны, их можно передать вышележащему протоколу, например TCP.

Выводы

Протокол IP решает задачу доставки сообщений между узлами составной сети. Поскольку он является дейтаграммным, никаких гарантий надежной доставки сообщений не дается.

Максимальная длина IP-пакета составляет 65 535 байт. Заголовок обычно имеет длину 20 байт и содержит информацию о сетевых адресах отправителя и получателя, параметры фрагментации, время жизни пакета, контрольную сумму и некоторые другие параметры.

Вид таблицы IP-маршрутизации зависит от конкретной реализации маршрутизатора. Несмотря на значительные внешние различия выводимых на экран таблиц, все они включают два обязательных поля — это поля адреса назначения и следующего маршрутизатора.

Записи в таблицу маршрутизации могут поступать из разных источников. Во-первых, в результате конфигурирования программное обеспечение стека TCP/IP заносит в таблицу записи о непосредственно подключенных сетях и маршрутизаторах по умолчанию, а также записи об особых адресах. Во-вторых, администратор вручную заносит записи о специфических маршрутах и о маршруте по умолчанию. В-третьих, протоколы маршрутизации автоматически заносят в таблицу динамические записи об имеющихся маршрутах.

Эффективным средством структуризации IP-сетей являются маски. Маски позволяют разделить одну сеть на несколько подсетей или объединить несколько сетей в одну более крупную сеть.

Значительная роль в будущих IP-сетях отводится технологии бесклассовой междоменной маршрутизации (CIDR), которая решает две основные задачи. Первая состоит в более экономном расходовании адресного пространства, вторая — в уменьшении числа записей в таблицах.

Важной особенностью протокола IP, отличающей его от других сетевых протоколов, например от сетевого протокола IPX, является его способность выполнять динамическую фрагментацию пакетов при передаче их между сетями с различными максимально допустимыми значениями длины поля данных кадров (MTU).

Вопросы и задания

1. Сравните таблицу моста или коммутатора с таблицей маршрутизатора. Каким образом формируются эти таблицы? Какую информацию содержат? От чего зависит их объем?
2. Верно ли утверждение, что широковещательная рассылка является частным случаем групповой рассылки? Произвольной рассылки?
3. Может ли один сетевой интерфейс иметь одновременно несколько IPv6-адресов разных типов: уникальный адрес, адрес произвольной рассылки, групповой адрес?
4. Рассмотрим маршрутизатор на магистрали Интернета. Какие записи содержатся в поле адреса назначения его таблицы маршрутизации? Варианты ответов:

   а) номера всех сетей Интернета;

   б) номера некоторых сетей Интернета;

   в) номера некоторых сетей и адреса некоторых конечных узлов Интернета;

   г) номера сетей, подсоединенных к интерфейсам данного маршрутизатора.
5. Сколько записей о маршрутах по умолчанию может включать таблица маршрутизации?
6. Приведите примеры, когда может возникнуть необходимость в использовании специфических маршрутов?
7. Передается ли в IP-пакете маска в тех случаях, когда маршрутизация реализуется с использованием масок?
8. Какие преимущества дает технология CIDR? Что мешает ее широкому внедрению?
9. Пусть префикс непрерывного пула IP-адресов составляет 15 двоичных разрядов. Сколько адресов, входит в этот пул? Варианты ответов:

   а) 2^{15}; б) 2^{17}; в) $2^{15} - 2$; г) 15^2.

10. Почему в записи о маршруте по умолчанию в качестве адреса сети назначения часто указывается 0.0.0.0 с маской 0.0.0.0?
11. Какие элементы сети могут выполнять фрагментацию? Варианты ответов:
    а) только компьютеры;
    б) только маршрутизаторы;
    в) компьютеры, маршрутизаторы, мосты, коммутаторы;
    г) компьютеры и маршрутизаторы.
12. Что произойдет, если при передаче пакета он был фрагментирован и один из фрагментов не дошел до узла назначения после истечения тайм-аута? Варианты ответов:
    а) модуль IP получателя сообщит о неполучении одного фрагмента, а IP-модуль узла-отправителя повторит передачу недошедшего фрагмента;
    б) модуль IP получателя сообщит о неполучении одного фрагмента, а IP-модуль узла-отправителя повторит передачу всего пакета, в состав которого входил недошедший фрагмент;
    в) модуль IP узла-получателя отбросит все полученные фрагменты пакета, в котором потерялся один фрагмент, а IP-модуль узла-отправителя не будет предпринимать никаких действий по повторной передаче данного пакета.
13. Верно ли утверждение, что широковещательная рассылка является частным случаем групповой рассылки? Произвольной рассылки?
14. В разделе «Перекрытие адресных пространств» этой главы приведен пример того, как администратор планирует сеть своего предприятия. Решите ту же задачу по планированию сети, но для случая, когда для сети Ethernet требуется 300 адресов, для сети Token Ring — 30, для DMZ — 20 и для соединительной сети — 8. Какой пул адресов необходимо получить у поставщика услуг на этот раз? (Для определенности будем считать, что поставщик услуг выделит непрерывный пул адресов.) Как администратор распределит адреса между своими четырьмя сетями? Как будут выглядеть таблицы маршрутизации R1 и R2?

ГЛАВА 17 Базовые протоколы TCP/IP

Эту главу мы начнем с изучения протоколов TCP и UDP, исполняющих посредническую роль между приложениями и транспортной инфраструктурой сети. В то время как задачей уровня межсетевого взаимодействия, к которому относится протокол IP, является передача данных между сетевыми интерфейсами в составной сети, главная задача транспортного уровня, которую решают протоколы TCP и UDP, заключается в передаче данных между *прикладными процессами*, выполняющимися на компьютерах в сети.

Далее в этой главе рассматриваются протоколы маршрутизации, предназначенные для автоматического построения таблиц маршрутизации, на основе которых происходит продвижение пакетов сетевого уровня. Протоколы маршрутизации, в отличие от сетевых протоколов, таких как IP и IPX, не являются обязательными, так как таблица маршрутизации может строиться администратором сети вручную. Однако в крупных сетях со сложной топологией и большим количеством альтернативных маршрутов протоколы маршрутизации выполняют очень важную и полезную работу, автоматизируя построение таблиц маршрутизации, а также отыскивая новые маршруты при изменениях сети: отказах или появлении новых линий связи и маршрутизаторов.

Мы рассмотрим также протокол ICMP, являющийся средством оповещения отправителя о причинах недоставки его пакетов адресату. Помимо диагностики ICMP используется для *мониторинга* сети. Так, в основе популярных утилит мониторинга IP-сетей ping и traceroute лежат ICMP-сообщения.

Протоколы транспортного уровня TCP и UDP

К транспортному уровню стека TCP/IP относятся:

- **протокол управления передачей** (Transmission Control Protocol, TCP), описанный в стандарте RFC 793;
- **протокол пользовательских дейтаграмм** (User Datagram Protocol, UDP), описанный в стандарте RFC 768.

Протоколы TCP и UDP, как и протоколы прикладного уровня, устанавливаются на конечных узлах.

Порты и сокеты

В то время как задачей уровня сетевого взаимодействия, к которому относится протокол IP, является передача данных между сетевыми интерфейсами в составной сети, главная задача протоколов транспортного уровня TCP и UDP заключается в передаче данных между *прикладными процессами*, выполняющимися на компьютерах в сети.

Каждый компьютер может выполнять несколько процессов, более того, даже отдельный прикладной процесс может иметь несколько точек входа, выступающих в качестве адресов назначения для пакетов данных. Поэтому доставка данных на сетевой интерфейс компьютера-получателя — это еще не конец пути, так как данные необходимо переправить конкретному процессу-получателю. Процедура распределения протоколами TCP и UDP поступающих от сетевого уровня пакетов между прикладными процессами называется **демультиплексированием** (рис. 17.1).

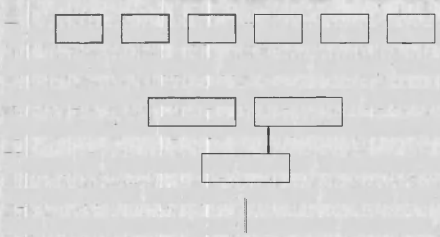
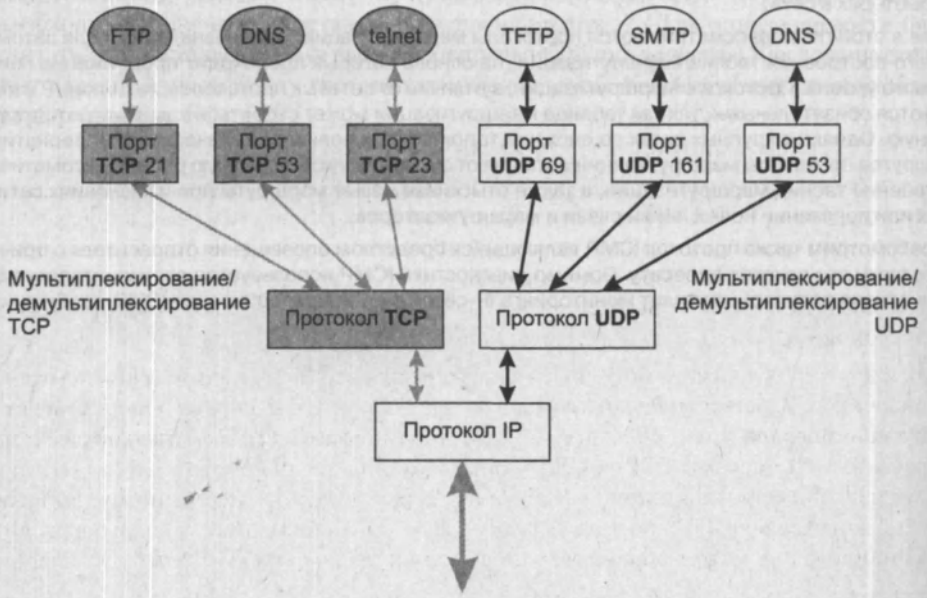


Рис. 17.1. Мультиплексирование и демультиплексирование на транспортном уровне



Существует и обратная задача: данные, генерируемые разными приложениями, работающими на одном конечном узле, должны быть переданы общему для всех них протокольному модулю IP для последующей отправки в сеть. Эту работу, называемую **мультиплексированием**, тоже выполняют протоколы TCP и UDP.

Протоколы TCP и UDP ведут для каждого приложения две системные очереди: очередь данных, поступающих к приложению из сети, и очередь данных, отправляемых этим приложением в сеть. Такие системные очереди называются **портами**[1], причем входная и выходная очереди одного приложения рассматриваются как один порт. Для идентификации портов им присваивают номера.

Если процессы представляют собой популярные системные службы, такие как FTP, telnet, HTTP, TFTP, DNS и т. п., то за ними закрепляются **стандартные назначенные номера**, называемые также **хорошо известными** (well-known) номерами портов. Эти номера закрепляются и публикуются в стандартах Интернета (RFC 1700, RFC 3232). Так, номер 21 закреплен за серверной частью службы удаленного доступа к файлам FTP, а 23 — за серверной частью службы удаленного управления telnet. Назначенные номера из диапазона от 0 до 1023 являются *уникальными в пределах Интернета* и закрепляются за приложениями *централизованно*.

Для тех приложений, которые еще не стали столь распространенными, номера портов назначаются *локально* разработчиками этих приложений или операционной системой в ответ на поступление запроса от приложения. На каждом компьютере операционная система ведет список занятых и свободных номеров портов. При поступлении запроса от приложения, выполняемого на данном компьютере, операционная система выделяет ему первый свободный номер. Такие номера называют **динамическими**. В дальнейшем все сетевые приложения должны адресоваться к данному приложению с указанием назначенного ему динамического номера порта. После того как приложение завершит работу, его номер возвращается в список свободных и может быть назначен другому приложению. Динамические номера являются *уникальными в пределах каждого компьютера*, но при этом обычной является ситуация совпадения номеров портов приложений, выполняемых на разных компьютерах. Как правило, клиентские части известных приложений (DNS, WWW, FTP, telnet и др.) получают динамические номера портов от ОС.

Все, что было сказано о портах, в равной степени относится к обоим протоколам транспортного уровня (TCP и UDP). В принципе, нет никакой зависимости между назначением номеров портов для приложений, использующих протокол TCP, и приложений, работающих с протоколом UDP. Приложения, которые передают данные на уровень IP по протоколу UDP, получают номера, называемые **UDP-портами**. Аналогично, приложениям, обращающимся к протоколу TCP, выделяются **TCP-порты**.

В том и другом случаях это могут быть как назначенные, так и динамические номера. Диапазоны чисел, из которых выделяются номера TCP- и UDP-портов, совпадают: от 0 до 1023 для назначенных и от 1024 до 65 535 для динамических. Однако никакой связи между назначенными номерами TCP- и UDP-портов нет. Даже если номера TCP- и UDP-портов совпадают, они идентифицируют разные приложения. Например, одному приложению может быть назначен TCP-порт 1750, а другому — UDP-порт 1750. В некоторых случаях, когда приложение может обращаться по выбору к протоколу TCP или UDP (например,

[1] Порт приложения не надо путать с портами (сетевыми интерфейсами) оборудования.

таким приложением является DNS), ему, исходя из удобства запоминания, назначаются совпадающие номера TCP- и UDP-портов (в данном примере — это *хорошо известный* номер 53).

Стандартные назначенные номера портов уникально идентифицируют тип приложения (FTP, или HTTP, или DNS и т. д.), однако они не могут использоваться для однозначной идентификации прикладных процессов, связанных с каждым из этих типов приложений. Пусть, например, на одном хосте запущены две *копии* DNS-сервера — DNS-сервер 1, DNS-сервер 2 (рис. 17.2). Каждый из этих DNS-серверов имеет хорошо известный UDP-порт 53. Какому из этих серверов нужно было бы направить запрос клиента, если бы в DNS-запросе в качестве идентификатора сервера был указан только номером порта?

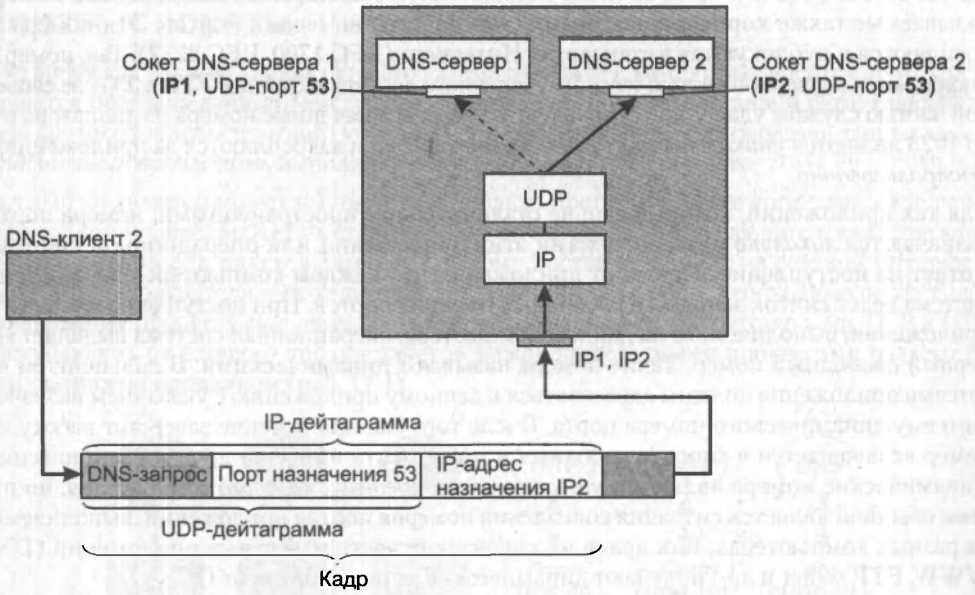


Рис. 17.2. Демультиплексирование протокола UDP на основе сокетов

Чтобы снять неоднозначность в идентификации приложений, разные копии связываются с разными IP-адресами. Для этого сетевой интерфейс компьютера, на котором выполняется несколько копий приложения, должен иметь соответствующее число IP-адресов — на рисунке это IP1 и IP2. Во всех IP-пакетах, направляемых DNS-серверу 1, в качестве IP-адреса указывается IP1, а DNS-серверу 2 — адрес IP2. Поэтому показанный на рисунке пакет, в поле данных которого содержится UDP-дейтаграмма с указанным номером порта 53, а в поле заголовка задан адрес IP2, буден направлен однозначно определенному адресату — DNS-серверу 2.

Прикладной процесс однозначно определяется в пределах сети и в пределах отдельного компьютера парой (IP-адрес, номер порта), называемой **сокетом** (socket). Сокет, определенный IP-адресом и номером UDP-порта, называется **UDP-сокетом**, а IP-адресом и номером TCP-порта — **TCP-сокетом**.

ПРИМЕЧАНИЕ

Здесь мы должны уточнить описанную в предыдущих главах упрощенную картину прохождения пакета вверх по стеку. Действительно, как мы и отмечали, после получения IP-пакета от протокола канального уровня протокол IP анализирует содержимое заголовка этого пакета, после чего заголовок отбрасывается, и «наверх» передается содержимое поля данных IP-пакета, например UDP-дейтаграмма. Упрощение состоит в том, что вместе с содержимым поля данных на транспортный уровень передается извлеченный из заголовка IP-адрес назначения, который и используется для однозначной идентификации приложения.

Протокол UDP и UDP-дейтаграммы

Протокол UDP, подобно IP, является дейтаграммным протоколом, реализующим так называемый ненадежный сервис *по возможности*, который не гарантирует доставку сообщений адресату.

При работе на хосте-отправителе данные от приложений поступают протоколу UDP через порт в виде сообщений (рис. 17.3). Протокол UDP добавляет к каждому отдельному сообщению свой 8-байтный заголовок, формируя из этих сообщений собственные протокольные единицы, называемые **UDP-дейтаграммами**, и передает их нижележащему протоколу IP. В этом и заключаются его функции по *мультиплексированию* данных.

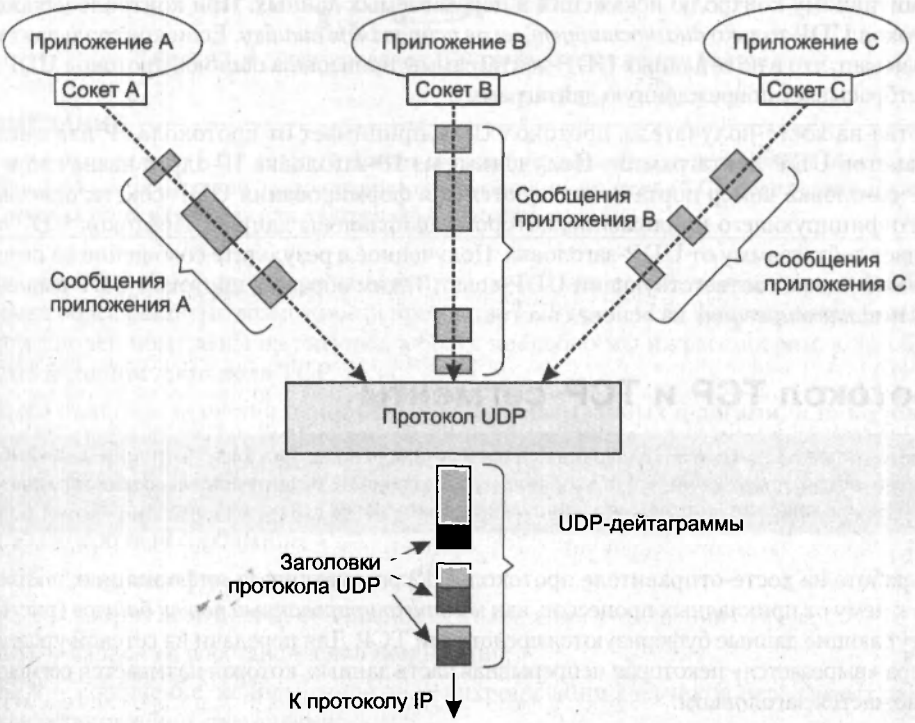


Рис. 17.3. Работа протокола UDP на хосте-отправителе

Каждая дейтаграмма переносит *отдельное пользовательское сообщение*. Сообщения могут иметь различную длину, не превышающую однако длину поля данных протокола IP, которое, в свою очередь, ограничено размером кадра технологии нижнего уровня. Поэтому если буфер UDP переполняется, то сообщение приложения отбрасывается.

Заголовок UDP состоит из четырех 2-байтных полей:

- номер UDP-порта отправителя;
- номер UDP-порта получателя;
- контрольная сумма;
- длина дейтаграммы.

Далее приведен пример заголовка UDP с заполненными полями:

```
Source Port = 0x0035
Destination Port = 0x0411
Total length = 132 (0x84) bytes
Checksum = 0x5333
```

В этой UDP-дейтаграмме в поле данных, длина которого, как следует из заголовка, равна (132 – 8) байт, помещено сообщение DNS-сервера, что можно видеть по номеру порта источника (`Source Port = 0–0035`). В шестнадцатеричном формате это значение равно стандартному номеру порта DNS-сервера — 53.

Судя по простоте заголовка, протокол UDP не сложен. Действительно, его функции сводятся к простой передаче данных между прикладным и сетевым уровнями, а также примитивному контролю искажений в передаваемых данных. При контроле искажений протокол UDP только *диагностирует, но не исправляет ошибку*. Если контрольная сумма показывает, что в поле данных UDP-дейтаграммы произошла ошибка, протокол UDP просто отбрасывает поврежденную дейтаграмму.

Работая на хосте-получателе, протокол UDP принимает от протокола IP извлеченные из пакетов UDP-дейтаграммы. Полученные из IP-заголовка IP-адрес назначения и из UDP-заголовка номер порта используются для формирования UDP-сокета, однозначно идентифицирующего приложение, которому направлены данные. Протокол UDP освобождает дейтаграмму от UDP-заголовка. Полученное в результате сообщение он передает приложению на соответствующий UDP-сокет. Таким образом, протокол UDP выполняет *демультиплексирование* на основе сокетов.

Протокол TCP и TCP-сегменты

Протокол TCP предназначен для передачи данных между приложениями. Этот протокол основан на *логическом соединении*, что позволяет ему обеспечивать гарантированную доставку данных, используя в качестве инструмента ненадежный дейтаграммный сервис протокола IP.

При работе на хосте-отправителе протокол TCP рассматривает информацию, поступающую к нему от прикладных процессов, как *неструктурированный поток байтов* (рис. 17.4). Поступающие данные буферизуются средствами TCP. Для передачи на сетевой уровень из буфера «вырезается» некоторая непрерывная часть данных, которая называется **сегментом**[1] и снабжается заголовком.

[1] Заметим, что сегментом называют как единицу передаваемых данных в целом (поле данных и заголовок протокола TCP), так и отдельно поле данных.

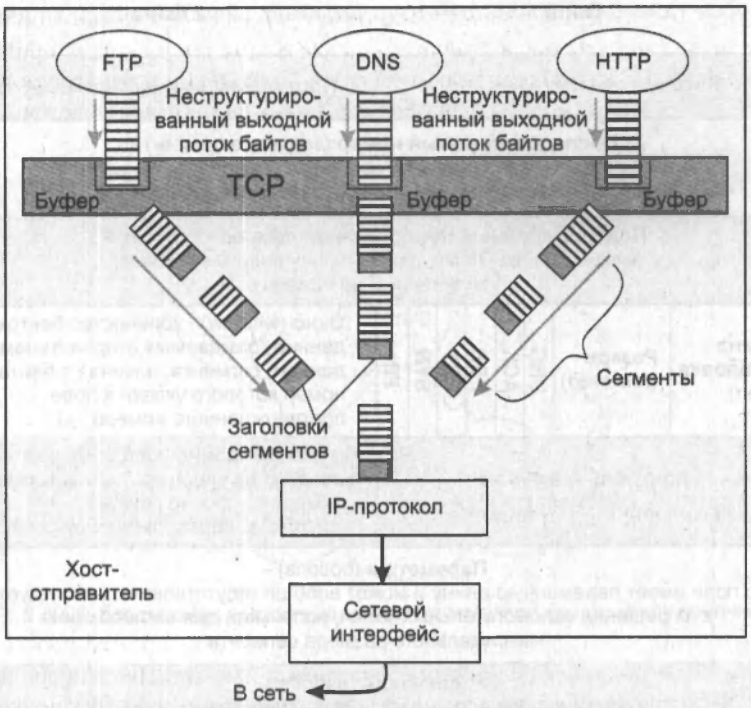


Рис. 17.4. Формирование TCP-сегментов из потока байтов

ПРИМЕЧАНИЕ

В отличие от протокола UDP, который создает свои дейтаграммы на основе логически обособленных единиц данных — сообщений, генерируемых приложениями, протокол TCP делит поток данных на сегменты без учета их смысла или внутренней структуры.

Заголовок TCP-сегмента содержит значительно больше полей, чем заголовок UDP, что отражает более развитые возможности протокола TCP (рис. 17.5). Краткие описания большинства полей помещены на рисунке, а более подробно мы их рассмотрим, когда будем изучать функции протокола TCP.

Коротко поясним значение однобитных полей, называемых **флагами**, или **кодовыми битами** (code bits). Они расположены сразу за резервным полем и содержат служебную информацию о типе данного сегмента. Положительное значение сигнализируется установкой этих битов в единицу:

- **URG** — срочное сообщение;
- **ACK** — квитанция на принятый сегмент;
- **PSH** — запрос на отправку сообщения без ожидания заполнения буфера;
- **RST** — запрос на восстановление соединения;
- **SYN** — сообщение, используемое для синхронизации счетчиков переданных данных при установлении соединения;
- **FIN** — признак достижения передающей стороной последнего байта в потоке передаваемых данных.

Рис. 17.5. Формат заголовка TCP-сегмента

Логические соединения — основа надежности TCP

Основным отличием TCP от UDP является то, что на протокол TCP возложена дополнительная задача — обеспечить *надежную доставку сообщений*, используя в качестве основы *ненадежный дейтаграммный протокол IP*.

Для решения этой задачи протокол TCP использует метод продвижения данных с установлением *логического соединения*. Как было сказано ранее, логическое соединение дает возможность участникам обмена следить за тем, чтобы данные не были потеряны, искажены или продублированы, а также чтобы они пришли к получателю в том порядке, в котором были отправлены.

Протокол TCP устанавливает логические соединения между *прикладными процессами*, причем в каждом соединении участвуют только *два* процесса. TCP-соединение является *дуплексным*, то есть каждый из участников этого соединения может одновременно получать и отправлять данные.

На рис. 17.6 показаны сети, соединенные маршрутизаторами, на которых установлен протокол IP. Установленные на конечных узлах протокольные модули TCP решают задачу

обеспечения надежного обмена данными путем установления между собой **логических соединений**.

Рис. 17.6. TCP-соединение создает надежный логический канал между конечными узлами

При установлении логического соединения модули TCP договариваются между собой о параметрах процедуры обмена данными. В протоколе TCP каждая сторона соединения посылает противоположной стороне следующие параметры:

- максимальный размер сегмента, который она готова принимать;
- максимальный объем данных (возможно несколько сегментов), которые она разрешает другой стороне передавать в свою сторону, даже если та еще не получила квитанцию на предыдущую порцию данных (размер окна);
- начальный порядковый номер байта, с которого она начинает отсчет потока данных в рамках данного соединения.

В результате переговорного процесса модулей TCP с двух сторон соединения определяются параметры соединения. Одни из них остаются постоянными в течение всего сеанса связи, а другие адаптивно изменяются. В частности, в зависимости от загрузки буфера принимающей стороны, а также надежности работы сети динамически изменяется размер окна отправителя.

Соединение устанавливается по инициативе клиентской части приложения. При необходимости выполнить обмен данными с серверной частью приложение-клиент обращается к нижележащему протоколу TCP, который в ответ на это обращение посылает сегмент-запрос на установление соединения протоколу TCP, работающему на стороне сервера (рис. 17.7, *а*). В числе прочего в запросе содержится флаг SYN, установленный в 1.

Получив запрос, модуль TCP на стороне сервера пытается создать «инфраструктуру» для обслуживания нового клиента. Он обращается к операционной системе с просьбой о выделении определенных системных ресурсов для организации буферов, таймеров, счетчиков. Эти ресурсы закрепляются за соединением с момента создания и до момента разрыва. Если на стороне сервера все необходимые ресурсы были получены и все необходимые действия выполнены, то модуль TCP посылает клиенту сегмент с флагами ACK и SYN.

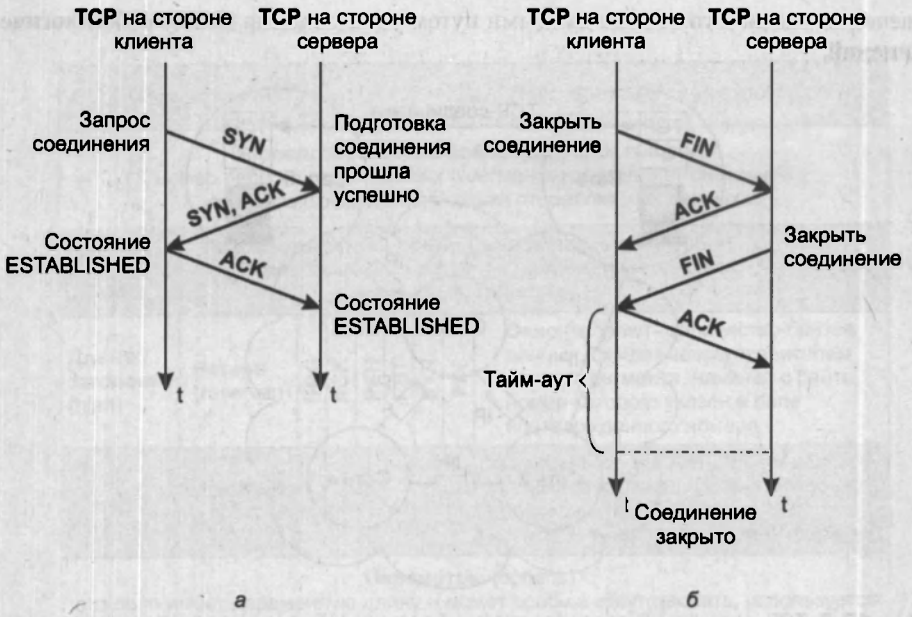


Рис. 17.7. Процедура установления и разрыва логического соединения при нормальном течении процесса

В ответ клиент посылает сегмент с флагом ACK и переходит в состояние установленного логического соединения (состояние ESTABLISHED). Когда сервер получает флаг ACK, он также переходит в состояние ESTABLISHED. На этом процедура установления соединения заканчивается, и стороны могут переходить к обмену данными.

Соединение может быть разорвано в любой момент по инициативе любой стороны. Для этого клиент и сервер должны обменяться сегментами FIN и ACK, в последовательности, показанной на рис. 17.7, *б* (здесь инициатором является клиент). Соединение считается закрытым по прошествии некоторого времени, в течение которого сторона-инициатор убеждается, что ее завершающий сигнал ACK дошел нормально и не вызвал никаких «аварийных» сообщений со стороны сервера.

ПРИМЕЧАНИЕ

Мы описали здесь процедуры установления и закрытия соединения очень схематично. Реальные протокольные модули работают в соответствии с гораздо более сложными алгоритмами, учитывающими всевозможные «нештатные» ситуации, такие, например, как задержки и потери сегментов, недостаточность ресурсов или неготовность сервера к установлению соединения. Кроме того, мы проигнорировали тот факт, что еще на этапе установления соединения стороны договариваются о некоторых параметрах своего взаимодействия, например о начальных номерах посылаемых ими байтов. Однако мы скоро вернемся к этим важным деталям работы протокола TCP.

Логическое TCP-соединение однозначно идентифицируется парой сокетов, определенных для этого соединения двумя взаимодействующими процессами.

Сокет одновременно может участвовать в нескольких соединениях. Так, на рис. 17.8 показаны три компьютера с адресами IP1, IP2, IP3. На каждом компьютере выполняется по

одному приложению — APPL1, APPL2 и APPL3, сокеты которых — соответственно (IP1, n1), (IP2, n2), (IP3, n3), а номера TCP-портов приложений — n1, n2, n3.

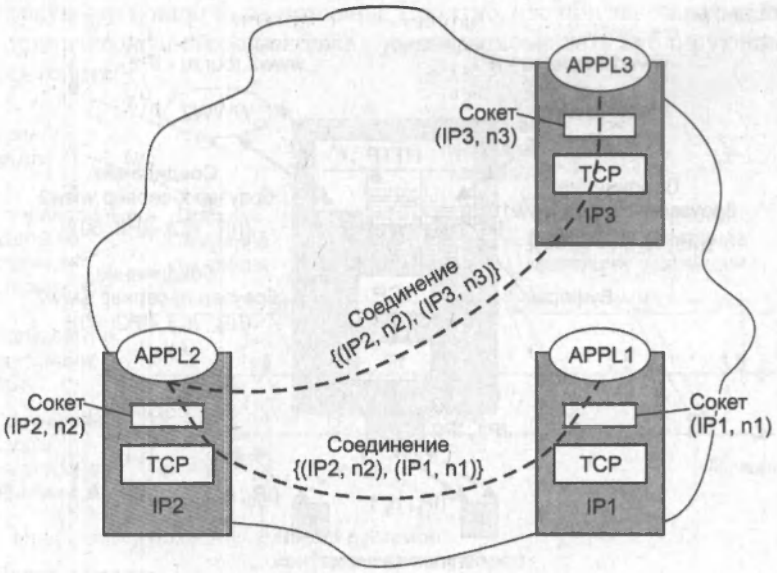


Рис. 17.8. Один сокет может участвовать в нескольких соединениях

На рисунке показаны два логических соединения, которое установило приложение 2 с приложением 1 и приложением 3. Логические соединения идентифицируются как {(IP2, n2), (IP1, n1)} и {(IP2, n2), (IP3, n3)} соответственно. Мы видим, что в обоих соединениях участвует один и тот же сокет — (IP2, n2).

А теперь рассмотрим на примере, как протокол TCP выполняет демультиплексирование. Пусть некий поставщик услуг оказывает услугу по веб-хостингу, то есть на его компьютере клиенты могут разворачивать свои веб-серверы. Веб-сервер основан на протоколе прикладного уровня HTTP, который передает свои сообщения в TCP-сегментах. Модуль TCP ожидает запросы от веб-клиентов (браузеров), «прослушивая» хорошо известный порт 80.

На рис. 17.9 показан вариант хостинга с двумя веб-серверами — сервером www1.model.ru, имеющим IP-адрес IP1, и сервером www2.tour.ru с адресом IP2. К каждому из них может обращаться множество клиентов, причем клиенты могут одновременно работать как с сервером www1, так и с сервером www2. Для каждой пары клиент-сервер протоколом TCP создается *отдельное логическое соединение*.

На рисунке показаны два браузера, имеющие соответственно сокеты (IP_k, n_k) и (IP_m, n_m). Пользователь браузера k обращается одновременно к серверам WWW1 и WWW2. Наличие отдельных соединений для работы с каждым из этих серверов обеспечивает не только надежную доставку, но и разделение информационных потоков — у пользователя никогда не возникает вопроса, каким сервером ему была послана та или иная страница. Одновременно с пользователем браузера k с сервером WWW2 работает пользователь браузера m. И в этом случае отдельные логические соединения, в рамках которых идет работа обоих пользователей, позволяют изолировать их информационные потоки. На рисунке показаны

буферы, количество которых определяется не числом веб-серверов и не числом клиентов, а числом логических соединений. Сообщения в эти буферы направляются в зависимости от значений сокетов как отправителя, так и получателя. Отсюда можно сделать вполне конкретный вывод.

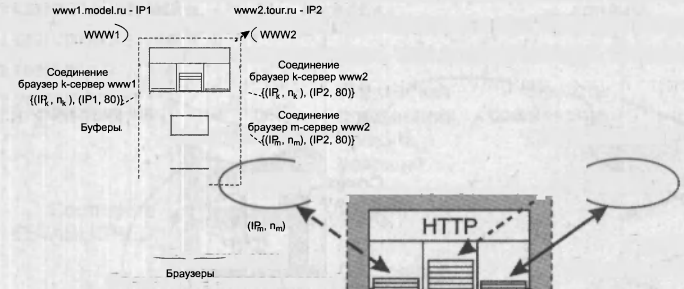


Рис. 17.9. Демультиплексирование протокола TCP на основе соединений

Протокол TCP осуществляет демультиплексирование информации, поступающей на прикладной уровень, на основе *соединений* процессов или, что одно и то же, на основе идентифицирующих эти процессы *пар сокетов*.

Повторная передача и скользящее окно

Один из наиболее естественных приемов, используемых для организации надежной передачи — это **квитирование**. Отправитель отсылает данные и ждет, пока к нему не придет квитанция, подтверждающая, что его данные благополучно дошли до адресата. В протоколе TCP используется частный случай квитирования — алгоритм скользящего окна. Прежде чем перейти к подробному рассмотрению особенностей реализации этого алгоритма в протоколе TCP, очень полезно обсудить его с общих позиций.

Итак, существует два метода организации процесса обмена квитанциями: метод простоя источника и метод скользящего окна.

Метод **простоя источника** требует, чтобы источник, пославший кадр (в данном случае не имеет значения, какое название используется для единицы передаваемых данных), дожидался от приемника квитанции, извещающей о том, что исходный кадр получен и данные в нем корректны, и только *после этого* посылал следующий кадр (или повторял искаженный). Если же квитанция в течение тайм-аута не пришла, то кадр (или квитанция) считается утерянным и его передача повторяется. На рис. 17.10 показано, что второй кадр

отсылается только после того, как пришла квитанция, подтверждающая доставку первого кадра. Однако затем произошла длительная пауза в отправке следующего третьего кадра. В течение этой паузы источник был вынужден повторить передачу кадра 2, так как квитанция на первую его копию была потеряна. Понятно, что при таком алгоритме работы источника принимающая сторона должна уметь распознавать дублирующиеся кадры и избавляться от них.

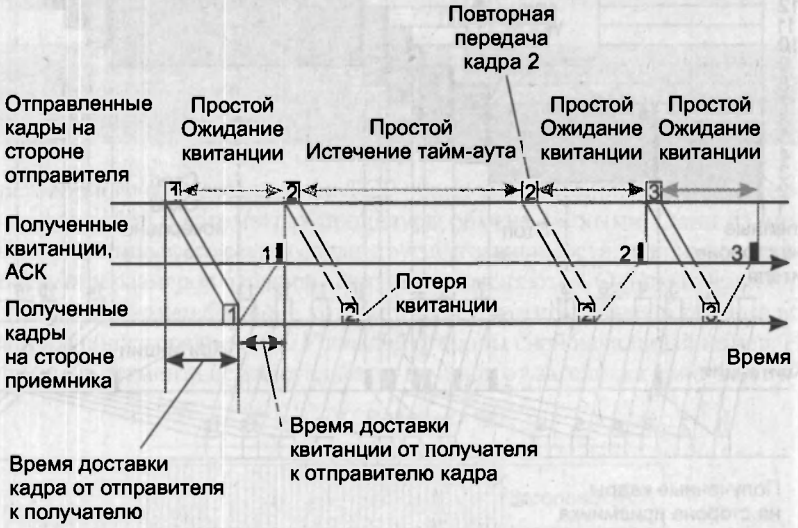


Рис. 17.10. Метод простоя источника

Достаточно очевидно, что при использовании данного метода производительность обмена данными ниже потенциально возможной — передатчик мог бы посылать следующий кадр сразу же после отправки предыдущего, но он обязан ждать прихода квитанции.

Второй метод называется методом **скользящего окна** (sliding window). В этом методе для повышения скорости передачи данных источнику разрешается передать некоторое количество кадров в непрерывном режиме, то есть в максимально возможном для источника темпе еще *до получения на эти кадры квитанций*. Количество кадров, которые разрешается передавать таким образом, называется **размером окна**.

Рисунок 17.11 иллюстрирует применение данного метода для окна размером 5 кадров.

В начальный момент, когда еще не послано ни одного кадра, окно определяет диапазон номеров кадров от 1 до 5 включительно. Источник начинает передавать кадры и через какое-то время получать в ответ квитанции. Для простоты предположим, что квитанции поступают в той же последовательности (но не обязательно в том же темпе), что и кадры, которым они соответствуют. В момент получения отправителем квитанции 1 окно сдвигается на одну позицию вверх, определяя новый диапазон разрешенных к отправке кадров (от 2 до 6).

Процессы отправки пакетов и получения квитанций идут достаточно независимо друг от друга. В нашем примере отправитель продолжает передавать кадры, но некоторое время не получает на них квитанции. После передачи кадра 6 окно исчерпывается, и источник приостанавливает передачу.

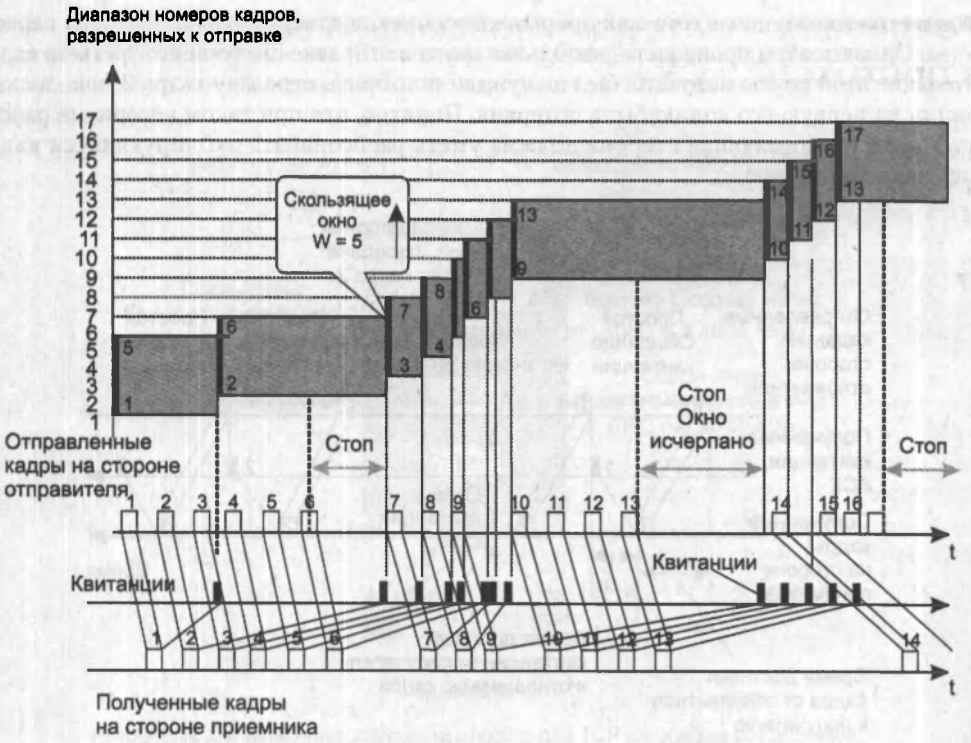


Рис. 17.11. Метод скользящего окна

После получения квитанции 2 (на кадр 2) окно сдвигается вверх на единицу, определяя диапазон разрешенных к передаче кадров от 3 до 7. Аналогичное «скольжение» окна вверх происходит *после получения каждой квитанции*: окно сдвигается вверх на 1, но его размер при этом не меняется и остается равным 5. После прихода квитанции 8 окно оказывается в диапазоне от 9 до 13 и остается таковым достаточно долго, так как по каким-то причинам источник перестает получать подтверждения о доставке кадров. Отправив последний разрешенный кадр 13, передатчик снова прекращает передачу с тем, чтобы возобновить ее после прихода квитанции 9.

При отправке кадра в источнике устанавливается тайм-аут. Если за установленное время квитанция на отправленный кадр не придет, то кадр (или квитанция на него) считается утерянным, и кадр передается снова. Если же поток квитанций поступает регулярно в пределах допуска в 5 кадров, то скорость обмена достигает максимально возможной величины для данного канала и принятого протокола.

В общем случае метод скользящего окна более сложен в реализации, чем метод простоя источника, так как передатчик должен хранить в буфере копии всех кадров, на которые пока не получены квитанции. Кроме того, при использовании данного метода требуется отслеживать несколько параметров алгоритма, таких как размер окна, номер кадра, на который получена квитанция, номер кадра, который еще можно передать до получения новой квитанции.

Реализация метода скользящего окна в протоколе TCP

Алгоритм скользящего окна в протоколе TCP имеет некоторые существенные особенности. В частности, в рассмотренном обобщенном алгоритме скользящего окна единицей передаваемых данных является кадр, и размер окна также определяется в кадрах, в то время как в протоколе TCP дело обстоит совсем по-другому.

Хотя единицей передаваемых данных протокола TCP является сегмент (аналог кадра в данном контексте), окно определено на множестве нумерованных *байтов* неструктурированного потока данных, передаваемого приложением протоколу TCP.

В ходе переговорного процесса модули TCP обоих участвующих в обмене сторон договариваются между собой о параметрах процедуры обмена данными. Одни из них остаются постоянными в течение всего сеанса связи, другие в зависимости, например, от интенсивности трафика и/или размеров буферов адаптивно изменяются. Одним из таких параметров является *начальный номер байта*, с которого будет вестись отсчет в течение всего функционирования данного соединения. У каждой стороны свой начальный номер. Нумерация байтов в пределах сегмента осуществляется, начиная от заголовка (рис. 17.12).

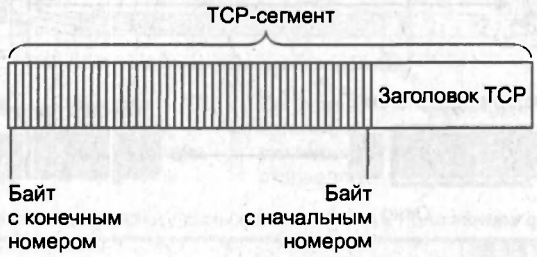


Рис. 17.12. Нумерация байтов в TCP-сегменте

Когда отправитель посылает TCP-сегмент, он помещает в поле *последовательного номера* номер первого байта данного сегмента, который служит *идентификатором* сегмента. На рис. 17.13 показаны четыре сегмента размером 1460 байт и один — 870 байт. Идентификаторами этих сегментов являются номера 32600, 34060, 35520 и т. д. На основании этих номеров получатель TCP-сегмента не только отличает данный сегмент от других, но и позиционирует полученный фрагмент относительно общего потока байтов. Кроме того, он может сделать вывод, например, что полученный сегмент является дубликатом или что между двумя полученными сегментами пропущены данные и т. д.

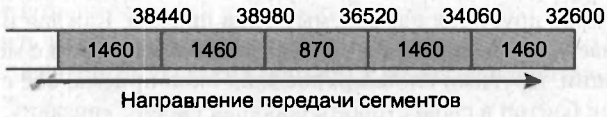


Рис. 17.13. Порядковый номер и номер квитанции

В качестве квитанции получатель сегмента отсылает ответное сообщение (сегмент), в поле *подтвержденного номера* которого он помещает число, на единицу превышающее макси-

мальный номер байта в полученном сегменте. Так, для первого отправленного сегмента, изображенного на рис. 17.13, квитанцией о получении (подтвержденным номером) будет число 34060, для второго — 35520 и т. д. Подтвержденный номер часто интерпретируют не только как оповещение о благополучной доставке, но и как номер следующего ожидаемого байта данных.

Квитанция в протоколе TCP посылается только в случае правильного приема данных. Таким образом, отсутствие квитанции означает либо потерю сегмента, либо потерю квитанции, либо прием искаженного сегмента.

В соответствии с определенным форматом один и тот же TCP-сегмент может нести в себе как пользовательские данные (в поле данных), так и квитанцию (в заголовке), которой подтверждается получение данных от другой стороны.

Поскольку протокол TCP является дуплексным, каждая сторона одновременно выступает и как отправитель, и как получатель. У каждой стороны есть пара буферов: один — для хранения принятых сегментов, другой — для сегментов, которые только еще предстоит отправить. Кроме того, имеется буфер для хранения копий сегментов, которые были отправлены, но квитанции о получении которых еще не поступили (рис. 17.14).

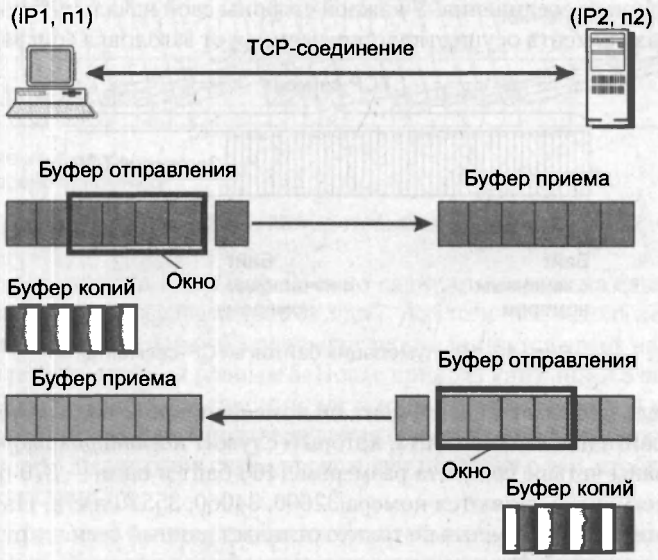


Рис. 17.14. Система буферов TCP-соединения

И при установлении соединения, и в ходе передачи обе стороны, выступая в роли получателя, посылают друг другу так называемые **окна приема**. Каждая из сторон, получив окно приема, «узнает», сколько байтов ей разрешается отправить с момента получения последней квитанции. Другими словами, посылая окна приема, обе стороны пытаются регулировать поток байтов в свою сторону, сообщая своему «визави», какое количество байтов (начиная с номера байта, о котором уже была выслана квитанция) они готовы в настоящий момент принять.

На рис. 17.15 показан поток байтов, поступающий от приложения в выходной буфер модуля TCP. Из потока байтов модуль TCP «нарезает» последовательность сегментов

и поочередно отправляет их приложению-получателю. Для определенности на рисунке принято направление перемещения данных справа налево. В этом потоке можно указать несколько логических границ:

- Первая граница отделяет сегменты, которые уже были отправлены и на которые уже пришли квитанции. Последняя квитанция пришла на байт с номером *N*.
- По другую сторону этой границы располагается окно размером *W* байт. Часть байтов, входящих в окно, составляют сегменты, которые также уже отправлены, но квитанции на которые пока не получены.
- Оставшаяся часть окна — это сегменты, которые пока не отправлены, но могут быть отправлены, так как входят в пределы окна.
- И наконец, последняя граница указывает на начало последовательности сегментов, ни один из которых не может быть отправлен до тех пор, пока не придет очередная квитанция и окно не будет сдвинуто вправо.

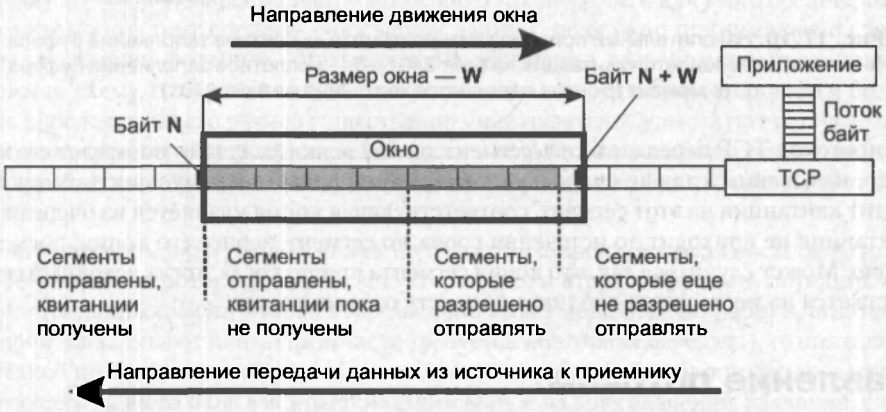


Рис. 17.15. Особенности реализации алгоритма скользящего окна в протоколе TCP

Если размер окна равен *W*, а последняя по времени квитанция содержала значение *N*, то отправитель может посылать новые сегменты до тех пор, пока в очередной сегмент не попадет байт с номером *N* + *W*. Этот сегмент выходит за рамки окна, и передачу в таком случае необходимо приостановить до прихода следующей квитанции.

Получатель может послать квитанцию, подтверждающую получение сразу нескольких сегментов, если они образуют непрерывный поток байтов. Например, (рис. 17.16, *а*), если в буфер, плотно без пропусков заполненный потоком байтов до 2354 включительно, поочередно поступили сегменты (2355–3816), (3817–5275) и (5276–8400), где цифры в скобках означают номера первых и последних байтов каждого сегмента, то получателю достаточно отправить только одну квитанцию на все три сегмента, указав в ней в качестве номера квитанции значение 8401. Таким образом, процесс квитирования в TCP является *накопительным*.

Вполне возможны ситуации, когда сегменты приходят к получателю не в том порядке, в котором были посланы, то есть в приемном буфере может образоваться «прогалина» (рис. 17.16, *б*). Пусть, к примеру, после указанных ранее трех сегментов вместо следующего по порядку сегмента (8401–10566) пришел сегмент (10567–12430). Очевидно, что

послать в качестве номера квитанции значение 12431 нельзя, потому что это бы означало, что получены все байты вплоть до 12430. Поскольку в потоке байтов образовался разрыв, получатель может только еще раз повторить квитанцию 8401, говоря тем самым, что все еще ожидает поступления потока байтов, начиная с 8401, то есть подтверждает получение не отдельных блоков данных, а непрерывной последовательности байтов.

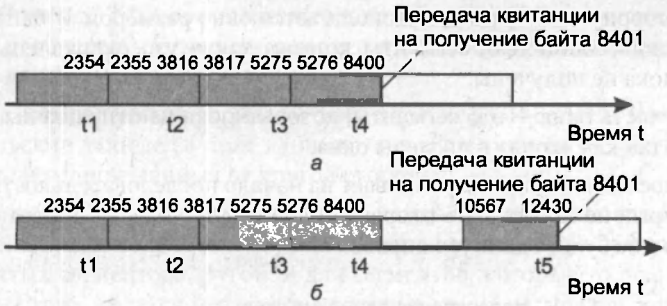


Рис. 17.16. Накопительный принцип квитирования: *а* — плотное заполнение буфера (в момент t4 передается квитанция на байт 8401), *б* — неплотное заполнение буфера (в момент t5 снова передается квитанция на байт 8401)

Когда протокол TCP передает в сеть сегмент, он «на всякий случай» помещает его копию в буфер, называемый также очередью повторной передачи, и запускает таймер. Когда приходит квитанция на этот сегмент, соответствующая копия удаляется из очереди. Если же квитанция не приходит до истечения срока, то сегмент, вернее его копия, посылается повторно. Может случиться так, что копия сегмента придет тогда, когда исходный сегмент уже окажется на месте, тогда дубликат попросту отбрасывается.

Управление потоком

Какой размер окна должен назначить источник приемнику, и наоборот? Точнее, каким на каждой из сторон должно быть выбрано время ожидания (тайм-аут) очередной квитанции? От ответа на этот вопрос зависит производительность протокола TCP.

При выборе величины *тайм-аута* должны учитываться скорость и надежность линий связи, их протяженность и многие другие факторы. Тайм-аут не должен быть слишком коротким, чтобы по возможности исключить избыточные повторные передачи, снижающие полезную пропускную способность системы, но он не должен быть и слишком длинным, чтобы избежать длительных простоев, связанных с ожиданием несуществующей или «заблудившейся» квитанции.

В протоколе TCP тайм-аут определяется с помощью достаточно сложного *адаптивного* алгоритма, идея которого состоит в следующем. При каждой передаче засекается время от момента отправки сегмента до прихода квитанции о его приеме (время оборота). Получаемые значения времени оборота усредняются с весовыми коэффициентами, возрастающими от предыдущего замера к последующему. Это делается с тем, чтобы усилить влияние последних замеров. В качестве тайм-аута выбирается среднее время оборота, умноженное на некоторый коэффициент. Практика показывает, что значение этого коэффициента должно превышать 2. В сетях с большим разбросом времени оборота при выборе тайм-аута учитывается и дисперсия этой величины.

Размер окна приема связан с наличием в данный момент места в буфере данных у принимающей стороны. Поэтому в общем случае окна приема на разных концах соединения имеют разный размер. Например, можно ожидать, что сервер, вероятно обладающий большим буфером, пошлет клиентской станции окно приема большее, чем клиент серверу. В зависимости от состояния сети то одна, то другая стороны могут объявлять новые значения окон приема, динамически уменьшая и увеличивая их.

Варьируя величину окна, можно влиять на загрузку сети. Чем больше окно, тем бóльшая порция неподтвержденных данных может быть послана в сеть. Но если пришло бóльшее количество данных, чем может быть принято модулем TCP, данные отбрасываются. Это ведет к излишним пересылкам информации и ненужному росту нагрузки на сеть и модуль TCP.

В то же время окно малого размера может ограничить передачу данных скоростью, которая определяется временем путешествия по сети каждого посылаемого сегмента. Чтобы избежать применения малых окон, в некоторых реализациях TCP предлагается получателю данных откладывать реальное изменение размеров окна до тех пор, пока свободное место не составит 20–40 % от максимально возможного объема памяти для этого соединения. Но и отправителю не стоит спешить с посылкой данных, пока окно принимающей стороны не станет достаточно большим. Учитывая эти соображения, разработчики протокола TCP предложили схему, согласно которой при установлении соединения заявляется большое окно, но впоследствии его размер существенно уменьшается. Существуют и другие прямо противоположные алгоритмы настройки окна, когда вначале выбирается минимальное окно, а затем, если сеть справляется с предложенной нагрузкой, его размер резко увеличивается.

Управлять размером окна приема может не только та сторона, которая посылает это окно, чтобы регулировать поток данных в свою сторону, но и вторая сторона — потенциальный отправитель данных. Если вторая сторона фиксирует ненадежную работу линии связи (регулярно запаздывают квитанции, часто требуется повторная передача), то она может по собственной инициативе уменьшить окно. В таких случаях действует правило: в качестве действующего размера окна выбирается минимальное из двух значений: значения, диктуемого приемной стороной, и значения, определяемого «на месте» отправителем.

Признаком перегрузки TCP-соединения является возникновение очередей на промежуточных узлах (маршрутизаторах) и на конечных узлах (компьютерах). При переполнении приемного буфера конечного узла «перегруженный» модуль TCP, отправляя квитанцию, помещает в нее новый уменьшенный размер окна. Если он совсем отказывается от приема, то в квитанции указывается *окно нулевого размера*. Однако даже после этого приложение может послать сообщение на отказавшийся от приема порт. Для этого сообщение должно сопровождаться *указателем срочности*. В такой ситуации порт обязан принять сегмент, даже если для этого придется вытеснить из буфера уже находящиеся там данные. После приема квитанции с нулевым значением окна протокол-отправитель время от времени делает контрольные попытки продолжить обмен данными. Если протокол-приемник уже готов принимать информацию, то в ответ на контрольный запрос он посылает квитанцию с указанием ненулевого размера окна.

Как видно из нашего далеко не полного описания двух протоколов транспортного уровня стека TCP/IP, на один из них — TCP — возложена сложная и очень важная задача: обеспечение надежной передачи данных через ненадежную сеть.

В то же время функциональная простота протокола UDP обуславливает простоту алгоритма его работы, компактность и высокое быстродействие. Поэтому те приложения,

в которых реализован собственный, достаточно надежный механизм обмена сообщениями, основанный на установлении соединения, предпочитают для непосредственной передачи данных по сети использовать менее надежные, но более быстрые средства транспортировки, в качестве которых по отношению к протоколу TCP и выступает протокол UDP. Протокол UDP может применяться и тогда, когда хорошее качество линий связи обеспечивает достаточный уровень надежности и без применения дополнительных приемов наподобие установления логического соединения и квитирования передаваемых пакетов. Заметим также, что поскольку протокол TCP основан на логических соединениях, он, в отличие от протокола UDP, *не годится для широковещательной и групповой рассылки*.

Общие свойства и классификация протоколов маршрутизации

Протоколы маршрутизации обеспечивают поиск и фиксацию маршрутов продвижения данных через составную сеть TCP/IP. Давайте остановимся на некоторых общих свойствах протоколов данного класса.

Начнем с того, что существуют такие способы продвижения пакетов в составных сетях, которые вообще *не требуют наличия таблиц маршрутизации на маршрутизаторах*.

Наиболее простым способом передачи пакетов по сети является так называемая **лавинная маршрутизация**, когда каждый маршрутизатор передает пакет всем своим непосредственным соседям, исключая тот, от которого его получил. Понятно, что это — не самый рациональный способ, так как пропускная способность сети используется крайне расточительно, тем не менее такой подход работоспособен (именно так мосты и коммутаторы локальных сетей поступают с кадрами, имеющими неизвестные адреса).

Еще одним видом маршрутизации, не требующим наличия таблиц маршрутизации, является **маршрутизация от источника** (source routing). В этом случае отправитель помещает в пакет информацию о том, какие промежуточные маршрутизаторы должны участвовать в передаче пакета к сети назначения. На основе этой информации каждый маршрутизатор считывает адрес следующего маршрутизатора, и если он действительно является адресом его непосредственного соседа, передает ему пакет для дальнейшей обработки. Вопрос о том, как отправитель узнает точный маршрут следования пакета через сеть, остается открытым. Маршрут может задавать либо вручную администратор, либо автоматически узел-отправитель, но в этом случае ему нужно поддерживать какой-либо протокол маршрутизации, который сообщит ему о топологии и состоянии сети. Маршрутизация от источника была опробована на этапе зарождения Интернета и сохранилась как практически неиспользуемая возможность протокола IPv4. В IPv6 маршрутизация от источника является одним из стандартных режимов продвижения пакетов, существует даже специальный заголовок для реализации этого режима.

Тем не менее большинство протоколов маршрутизации нацелено на *создание таблиц маршрутизации*.

Выбор рационального маршрута может осуществляться на основании различных *критериев*. Сегодня в IP-сетях применяются протоколы маршрутизации, в которых маршрут выбирается по критерию кратчайшего расстояния. При этом расстояние измеряется в различных метриках. Чаще всего используется простейшая метрика — количество хопов,

то есть количество маршрутизаторов, которые нужно преодолеть пакету до сети назначения. В качестве метрик применяются также пропускная способность и надежность каналов, вносимые ими задержки и любые комбинации этих метрик.

Различные протоколы маршрутизации обладают *разным временем конвергенции*.

Протокол маршрутизации должен обеспечить создание на маршрутизаторах *согласованных* друг с другом таблиц маршрутизации, то есть таких таблиц, которые обеспечат доставку пакета от исходной сети в сеть назначения за конечное число шагов. Современные протоколы маршрутизации поддерживают согласованность таблиц, однако это их свойство не абсолютно — при изменениях в сети, например при отказе каналов передачи данных или самих маршрутизаторов, возникают периоды нестабильной работы сети, вызванной временной несогласованностью таблиц разных маршрутизаторов. Протоколу маршрутизации обычно нужно некоторое время, которое называется **временем конвергенции**, чтобы после нескольких итераций обмена служебной информацией все маршрутизаторы сети внесли изменения в свои таблицы и в результате таблицы снова стали согласованными.

Различают протоколы, выполняющие статическую и адаптивную (динамическую) маршрутизацию.

При **статической маршрутизации** все записи в таблице имеют неизменяемый, статический статус, что подразумевает бесконечный срок их жизни. Записи о маршрутах составляются и вводятся в память каждого маршрутизатора *вручную администратором сети*. При изменении состояния сети администратору необходимо срочно отразить эти изменения в соответствующих таблицах маршрутизации, иначе может произойти их рассогласование, и сеть будет работать некорректно.

При **адаптивной маршрутизации** все изменения конфигурации сети *автоматически* отражаются в таблицах маршрутизации благодаря *протоколам маршрутизации*. Эти протоколы собирают информацию о топологии связей в сети, что позволяет им оперативно отрабатывать все текущие изменения. В таблицах маршрутизации при адаптивной маршрутизации обычно имеется информация об интервале времени, в течение которого данный маршрут будет оставаться действительным. Это время называют *временем жизни* (TTL) маршрута. Если по истечении времени жизни существование маршрута не подтверждается протоколом маршрутизации, то он считается нерабочим, пакеты по нему больше не посылаются.

Протоколы адаптивной маршрутизации бывают распределенными и централизованными.

При *распределенном* подходе все маршрутизаторы сети находятся в равных условиях, они находят маршруты и строят собственные таблицы маршрутизации, работая в тесной кооперации друг с другом, постоянно обмениваясь информацией о конфигурации сети. При *централизованном* подходе в сети существует один выделенный маршрутизатор, который собирает всю информацию о топологии и состоянии сети от других маршрутизаторов. На основании этих данных выделенный маршрутизатор (который иногда называют *сервером маршрутов*) строит таблицы маршрутизации для всех остальных маршрутизаторов сети, а затем распространяет их по сети, чтобы каждый маршрутизатор получил собственную таблицу и в дальнейшем самостоятельно принимал решение о продвижении каждого пакета.

Применяемые сегодня в IP-сетях протоколы маршрутизации относятся к *адаптивным распределенным* протоколам, которые, в свою очередь, делятся на две группы:

- дистанционно-векторные алгоритмы (Distance Vector Algorithm, DVA);
- алгоритмы состояния связей (Link State Algorithm, LSA).

В **дистанционно-векторных алгоритмах** (DVA) каждый маршрутизатор *периодически* и *широковещательно* рассылает по сети вектор, компонентами которого являются расстояния (измеренные в той или иной метрике) от данного маршрутизатора до всех известных ему сетей. Пакеты протоколов маршрутизации обычно называют *объявлениями о расстояниях*, так как с их помощью маршрутизатор объявляет остальным маршрутизаторам известные ему сведения о конфигурации сети.

Получив от некоторого соседа вектор расстояний (дистанций) до известных тому сетей, маршрутизатор наращивает компоненты вектора на величину расстояния от себя до данного соседа. Кроме того, он дополняет вектор информацией об известных ему самому других сетях, о которых он узнал непосредственно (если они подключены к его портам) или из аналогичных объявлений других маршрутизаторов. Обновленное значение вектора маршрутизатор рассылает своим соседям. В конце концов, каждый маршрутизатор узнает через соседние маршрутизаторы информацию обо всех имеющихся в составной сети сетях и о расстояниях до них.

Затем он выбирает из нескольких альтернативных маршрутов к каждой сети тот маршрут, который обладает наименьшим значением метрики. Маршрутизатор, передавший информацию о данном маршруте, отмечается в таблице маршрутизации как *следующий* (next hop).

Дистанционно-векторные алгоритмы хорошо работают только в небольших сетях. В больших сетях они периодически засоряют линии связи интенсивным трафиком, к тому же изменения конфигурации не всегда корректно могут отрабатываться алгоритмом этого типа, так как маршрутизаторы не имеют точного представления о топологии связей в сети, а располагают только косвенной информацией — вектором расстояний.

Наиболее распространенным протоколом, основанным на дистанционно-векторном алгоритме, является протокол RIP (см. далее).

Алгоритмы состояния связей (LSA) обеспечивают каждый маршрутизатор информацией, достаточной для построения точного графа связей сети. Все маршрутизаторы работают на основании одного и того же графа, что делает процесс маршрутизации более устойчивым к изменениям конфигурации.

Каждый маршрутизатор использует граф сети для нахождения оптимальных по некоторому критерию маршрутов до каждой из сетей, входящих в составную сеть.

Чтобы понять, в каком состоянии находятся линии связи, подключенные к его портам, маршрутизатор периодически обменивается короткими пакетами HELLO со своими ближайшими соседями. В отличие от протоколов DVA, которые регулярно передают вектор расстояний, протоколы LSA ограничиваются короткими сообщениями, а передача более объемных сообщений происходит только в тех случаях, когда с помощью сообщений HELLO был установлен факт изменения состояния какой-либо связи.

В результате служебный трафик, создаваемый протоколами LSA, гораздо менее интенсивный, чем у протоколов DVA.

Протоколами, основанными на алгоритме состояния связей, являются протокол IS-IS стека OSI (этот протокол используется также в стеке TCP/IP) и протокол OSPF стека TCP/IP.

Протокол RIP

Протокол RIP (Routing Information Protocol — протокол маршрутной информации) является внутренним протоколом маршрутизации дистанционно-векторного типа.

Будучи простым в реализации, этот протокол чаще всего используется в небольших сетях. Для IP имеются две версии RIP — RIPv1 и RIPv2. Протокол RIPv1 не поддерживает масок. Протокол RIPv2 передает информацию о масках сетей, поэтому он в большей степени соответствует требованиям сегодняшнего дня. Так как построение таблиц маршрутизации в обеих версиях протокола принципиально не отличается, в дальнейшем для упрощения записей будет описываться работа версии 1.

Построение таблицы маршрутизации

Для измерения расстояния до сети стандарты протокола RIP допускают различные виды метрик: хопы, значения пропускной способности, вносимые задержки, надежность сетей (то есть соответствующие признакам D, T и R в поле качества сервиса IP-пакета), а также любые комбинации этих метрик. Метрика должна обладать свойством *аддитивности* — метрика составного пути должна быть равна сумме метрик составляющих этого пути. В большинстве реализаций RIP используется простейшая метрика — количество хопов, то есть количество промежуточных маршрутизаторов, которые нужно преодолеть пакету до сети назначения.

Рассмотрим процесс построения таблицы маршрутизации с помощью протокола RIP на примере составной сети, изображенной на рис. 17.17. Мы разделим этот процесс на 5 этапов.

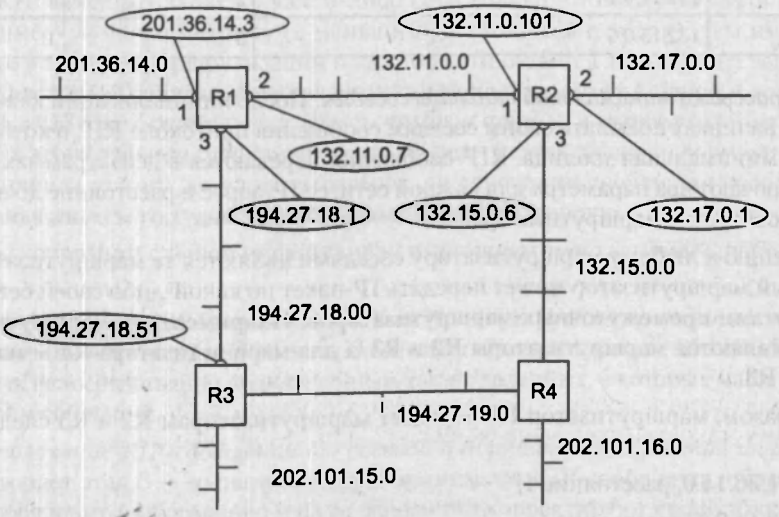


Рис. 17.17. Сеть, построенная на маршрутизаторах RIP

Этап 1 — создание минимальной таблицы. Данная составная сеть включает восемь IP-сетей, связанных четырьмя маршрутизаторами с идентификаторами: R1, R2, R3 и R4. Маршрутизаторы, работающие по протоколу RIP, могут иметь идентификаторы, однако

для протокола они не являются необходимыми. В RIP-сообщениях эти идентификаторы не передаются.

В исходном состоянии на каждом маршрутизаторе программным обеспечением стека TCP/IP автоматически создается минимальная таблица маршрутизации, в которой учитываются только непосредственно подсоединенные сети. На рисунке адреса портов маршрутизаторов в отличие от адресов сетей помещены в овалы.

Таблица 17.1 позволяет оценить примерный вид минимальной таблицы маршрутизации маршрутизатора R1.

Таблица 17.1. Минимальная таблица маршрутизации маршрутизатора R1

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	201.36.14.3	1	1
132.11.0.0	132.11.0.7	2	1
194.27.18.0	194.27.18.1	3	1

Минимальные таблицы маршрутизации в других маршрутизаторах будут выглядеть соответственно, например, таблица маршрутизатора R2 будет состоять из трех записей (табл. 17.2).

Таблица 17.2. Минимальная таблица маршрутизации маршрутизатора R2

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
132.11.0.0	132.11.0.101	1	1
132.17.0.0	132.17.0.1	2	1
132.15.0.0	132.15.0.6	3	1

Этап 2 — рассылка минимальной таблицы соседям. После инициализации каждый маршрутизатор начинает посылать своим соседям сообщения протокола RIP, в которых содержится его минимальная таблица. RIP-сообщения передаются в дейтаграммах протокола UDP и включают два параметра для каждой сети: ее IP-адрес и расстояние до нее от передающего сообщение маршрутизатора.

По отношению к любому маршрутизатору соседями являются те маршрутизаторы, которым данный маршрутизатор может передать IP-пакет по какой-либо своей сети, не пользуясь услугами промежуточных маршрутизаторов. Например, для маршрутизатора R1 соседями являются маршрутизаторы R2 и R3, а для маршрутизатора R4 — маршрутизаторы R2 и R3.

Таким образом, маршрутизатор R1 передает маршрутизаторам R2 и R3 следующие сообщения:

- сеть 201.36.14.0, расстояние 1;
- сеть 132.11.0.0, расстояние 1;
- сеть 194.27.18.0, расстояние 1.

Этап 3 — получение RIP-сообщений от соседей и обработка полученной информации. После получения аналогичных сообщений от маршрутизаторов R2 и R3 маршрутизатор R1 наращивает каждое полученное поле метрики на единицу и запоминает, через какой порт

и от какого маршрутизатора получена новая информация (адрес этого маршрутизатора станет адресом следующего маршрутизатора, если эта запись будет внесена в таблицу маршрутизации). Затем маршрутизатор начинает сравнивать новую информацию с той, которая хранится в его таблице маршрутизации (табл. 17.3).

Таблица 17.3. Таблица маршрутизации маршрутизатора R1

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	201.36.14.3	1	1
132.11.0.0	132.11.0.7	2	1
194.27.18.0	194.27.18.1	3	1
132.17.0.0	132.11.0.101	2	2
132.15.0.0	132.11.0.101	2	2
194.27.19.0	194.27.18.51	3	2
202.101.15.0	194.27.18.51	3	2
~~132.11.0.0~~	~~132.11.0.101~~	~~2~~	~~2~~
~~194.27.18.0~~	~~194.27.18.51~~	~~3~~	~~2~~

Записи с четвертой по девятую получены от соседних маршрутизаторов, и они претендуют на помещение в таблицу. Однако только записи с четвертой по седьмую попадают в таблицу, а записи восьмая и девятая — нет. Это происходит потому, что они содержат данные об уже имеющихся в таблице маршрутизатора R1 сетях, а расстояние до них больше, чем в существующих записях.

Протокол RIP замещает запись о какой-либо сети только в том случае, если новая информация имеет лучшую метрику (с меньшим расстоянием в хопах), чем имеющаяся. В результате в таблице маршрутизации о каждой сети остается только одна запись; если же имеется несколько записей, равнозначных в отношении путей к одной и той же сети, то все равно в таблице остается одна запись, которая пришла в маршрутизатор первая по времени. Для этого правила существует исключение — если худшая информация о какой-либо сети пришла от того же маршрутизатора, на основании сообщения которого была создана данная запись, то худшая информация замещает лучшую.

Аналогичные операции с новой информацией выполняют и остальные маршрутизаторы сети.

Этап 4 — рассылка новой таблицы соседям. Каждый маршрутизатор отсылает новое RIP-сообщение всем своим соседям. В этом сообщении он помещает данные обо всех известных ему сетях: как непосредственно подключенных, так и удаленных, о которых маршрутизатор узнал из RIP-сообщений.

Этап 5 — получение RIP-сообщений от соседей и обработка полученной информации. Этап 5 повторяет этап 3 — маршрутизаторы принимают RIP-сообщения, обрабатывают содержащуюся в них информацию и на ее основании корректируют свои таблицы маршрутизации.

Посмотрим, как это делает маршрутизатор R1 (табл. 17.4).

На этом этапе маршрутизатор R1 получает от маршрутизатора R3 информацию о сети 132.15.0.0, которую тот, в свою очередь, на предыдущем цикле работы получил от маршру-

тизатора R4. Маршрутизатор уже знает о сети 132.15.0.0, причем старая информация имеет лучшую метрику, чем новая, поэтому новая информация об этой сети отбрасывается.

Таблица 17.4. Таблица маршрутизации маршрутизатора R1

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	201.36.14.3	1	1
132.11.0.0	132.11.0.7	2	1
194.27.18.0	194.27.18.1	3	1
132.17.0.0	132.11.0.101	2	2
132.15.0.0	132.11.0.101	2	2
~~132.15.0.0~~	~~194.27.18.51~~	~~3~~	~~3~~
194.27.19.0	194.27.18.51	3	2
~~194.27.19.0~~	~~132.11.0.101~~	~~2~~	~~3~~
~~202.101.15.0~~	~~194.27.18.51~~	~~3~~	~~2~~
202.101.16.0	132.11.0.101	2	3
~~202.101.16.0~~	~~194.27.18.51~~	~~3~~	~~3~~

О сети 202.101.16.0 маршрутизатор R1 узнает на этом этапе впервые, причем данные о ней приходят от двух соседей — от R3 и R4. Поскольку метрики в этих сообщениях указаны одинаковые, то в таблицу попадают данные, пришедшие первыми. В нашем примере считается, что маршрутизатор R2 опередил маршрутизатор R3 и первым переслал свое RIP-сообщение маршрутизатору R1.

Если маршрутизаторы периодически повторяют этапы рассылки и обработки RIP-сообщений, то за конечное время в сети установится корректный режим маршрутизации. Под корректным режимом маршрутизации здесь понимается такое состояние таблиц маршрутизации, когда все сети достижимы из любой сети с помощью некоторого рационального маршрута. Пакеты будут доходить до адресатов и не зацикливаться в петлях, подобных той, которая образуется на рис. 17.17, маршрутизаторами R1, R2, R3 и R4.

Очевидно, если в сети все маршрутизаторы, их интерфейсы и соединяющие их линии связи остаются работоспособными, то объявления по протоколу RIP можно делать достаточно редко, например один раз в день. Однако в сетях постоянно происходят изменения — меняется работоспособность маршрутизаторов и линий связи, кроме того, маршрутизаторы и линии связи могут добавляться в существующую сеть или же выводиться из ее состава.

Для адаптации к изменениям в сети протокол RIP использует ряд механизмов.

Адаптация маршрутизаторов RIP к изменениям состояния сети

К новым маршрутам маршрутизаторы RIP приспосабливаются просто — они передают новую информацию в очередном сообщении своим соседям и постепенно эта информация становится известна всем маршрутизаторам сети. А вот к изменениям, связанным с потерей

какого-либо маршрута, маршрутизаторы RIP адаптируются сложнее. Это связано с тем, что в формате сообщений протокола RIP нет поля, которое бы указывало на то, что путь к данной сети больше не существует.

Для уведомления о том, что некоторый маршрут недействителен, используются два механизма:

- истечение времени жизни маршрута;
- указание специального (бесконечного) расстояния до сети, ставшей недоступной.

Механизм **истечения времени жизни маршрута** основан на том, что каждая запись таблицы маршрутизации (как и записи таблицы продвижения моста/коммутатора), полученная по протоколу RIP, имеет время жизни (TTL). При поступлении очередного RIP-сообщения, которое подтверждает справедливость данной записи, таймер времени жизни устанавливается в исходное состояние, а затем из него каждую секунду вычитается единица. Если за время тайм-аута не придет новое сообщение об этом маршруте, он помечается как *недействительный*.

Время тайм-аута связано с периодом рассылки векторов по сети. В протоколе RIP период рассылки выбран равным 30 секундам, а в качестве тайм-аута выбрано шестикратное значение периода рассылки, то есть 180 секунд. Шестикратный запас времени нужен для уверенности в том, что сеть действительно стала недоступной, а не просто произошли потери RIP-сообщений (а это возможно, так как протокол RIP использует транспортный протокол UDP, который не обеспечивает надежной доставки сообщений). Если какой-либо маршрутизатор отказывает, переставая слать своим соседям сообщения о сетях, которые можно достичь через него, то через 180 секунд все записи, порожденные этим маршрутизатором, у его ближайших соседей станут недействительными. После этого процесс повторится уже для ближайших соседей — они вычеркнут подобные записи уже через 360 секунд.

Как видно, сведения о сетях, пути к которым не могут теперь проходить через отказавший маршрутизатор, распространяются по сети не очень быстро. В этом заключается одна из причин выбора в качестве периода рассылки небольшой величины в 30 секунд. Механизм **тайм-аута** работает в тех случаях, когда маршрутизатор не может послать соседям сообщение об отказавшем маршруте, так как либо он сам неработоспособен, либо неработоспособна линия связи, по которой можно было бы передать сообщение.

Когда же сообщение послать можно, маршрутизаторы RIP используют прием, заключающийся в *указании бесконечного расстояния до сети, ставшей недоступной*. В протоколе RIP бесконечным условно считается расстояние в 16 хопов. Получив сообщение, в котором расстояние до некоторой сети равно 16 (или 15, что приводит к тому же результату, так как маршрутизатор наращивает полученное значение на 1), маршрутизатор должен проверить, исходит ли эта «плохая» информация о сети от того же маршрутизатора, сообщение которого послужило в свое время основанием для записи о данной сети в таблице маршрутизации. Если это тот же маршрутизатор, то информация считается достоверной и маршрут помечается как недоступный.

Причиной выбора в качестве «бесконечного» расстояния столь небольшого числа является то, что в некоторых случаях отказы связей в сети вызывают длительные периоды некорректной работы маршрутизаторов RIP, выражающейся в зацикливании пакетов в петлях сети. И чем меньше расстояние, используемое в качестве «бесконечного», тем такие периоды короче.

Пример зацикливания пакетов

Рассмотрим случай зацикливания пакетов на примере сети, изображенной на рис. 17.17. Пусть маршрутизатор R1 обнаружил, что его связь с непосредственно подключенной сетью 201.36.14.0 потеряна (например, по причине отказа интерфейса 201.36.14.3). Маршрутизатор R1 отмечает в своей таблице маршрутизации, что сеть 201.36.14.0 недоступна. В худшем случае он обнаружит это сразу же после отправки очередных RIP-сообщений, так что до начала нового цикла его объявлений, в котором он должен сообщить соседям, что расстояние до сети 201.36.14.0 стало равным 16, остается почти 30 секунд. Каждый маршрутизатор работает на основании своего внутреннего таймера, не синхронизируя работу по рассылке объявлений с другими маршрутизаторами. Поэтому весьма вероятно, маршрутизатор R2 опередит маршрутизатор R1 и передаст ему свое сообщение раньше, чем R1 успеет передать новость о недостижимости сети 201.36.14.0. А в этом сообщении имеются данные, порожденные записью в таблице маршрутизации R2 (табл. 17.5).

Таблица 17.5. Таблица маршрутизации маршрутизатора R2

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	132.11.0.7	1	2

Эта запись, полученная от маршрутизатора R1, была корректна до отказа интерфейса 201.36.14.3; теперь она устарела, но маршрутизатор R2 об этом не знает.

Далее маршрутизатор R1 получает новую информацию о сети 201.36.14.0 — эта сеть достижима через маршрутизатор R2 с метрикой 2. Раньше R1 также получал эту информацию от R2, но игнорировал ее, так как его собственная метрика для 201.36.14.0 была лучше. Теперь R1 должен принять данные о сети 201.36.14.0, полученные от R2, и заменить запись в таблице маршрутизации о недостижимости этой сети (табл. 17.6).

Таблица 17.6. Таблица маршрутизации маршрутизатора R1

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	132.11.0.101	2	3

В результате в сети образуется маршрутная петля: пакеты, направляемые узлам сети 201.36.14.0, станут передаваться маршрутизатором R2 маршрутизатору R1, а маршрутизатор R1 будет возвращать их маршрутизатору R2. IP-пакеты продолжат циркулировать по этой петле до тех пор, пока не истечет время жизни каждого пакета. Рассмотрим периоды времени, кратные времени жизни записей в таблицах маршрутизаторов.

- *Время 0–180 с.* После отказа интерфейса в маршрутизаторах R1 и R2 будут сохраняться некорректные записи. Маршрутизатор R2 по-прежнему снабжает маршрутизатор R1 своей записью о сети 201.36.14.0 с метрикой 2, так как ее время жизни не истекло. Пакеты зацикливаются.
- *Время 180–360 с.* В начале этого периода у маршрутизатора R2 истекает время жизни записи о сети 201.36.14.0 с метрикой 2, так как маршрутизатор R1 в предыдущий период посылал ему сообщения о сети 201.36.14.0 с худшей метрикой, чем у R2, и они не могли подтверждать эту запись. Теперь маршрутизатор R2 принимает от маршрутизатора R1 запись о сети 201.36.14.0 с метрикой 3 и трансформирует ее в запись

с метрикой 4. Маршрутизатор R1 не получает новых сообщений от маршрутизатора R2 о сети 201.36.14.0 с метрикой 2, поэтому время жизни его записи начинает уменьшаться. Пакеты продолжают зацикливаться.

- *Время 360–540 с.* У маршрутизатора R1 истекает время жизни записи о сети 201.36.14.0 с метрикой 3. Маршрутизаторы R1 и R2 опять меняются ролями — R2 снабжает R1 устаревшей информацией о пути к сети 201.36.14.0, уже с метрикой 4, которую R1 преобразует в метрику 5. Пакеты продолжают зацикливаться.

Если бы в протоколе RIP не было выбрано расстояние 16 в качестве недостижимого, то описанный процесс длился бы бесконечно (вернее, пока не была бы исчерпана разрядная сетка поля расстояния, и при очередном наращивании расстояния было бы зафиксировано переполнение).

В результате маршрутизатор R2 на очередном этапе описанного процесса получает от маршрутизатора R1 метрику 15, которая после наращивания, превращаясь в метрику 16, фиксирует недостижимость сети. Таким образом, в нашем примере период нестабильной работы сети длился 36 минут!

Ограничение в 15 хопов сужает область применения протокола RIP до сетей, в которых число промежуточных маршрутизаторов не может быть больше 15. Для более масштабных сетей нужно применять другие протоколы маршрутизации, например OSPF, или разбивать сеть на автономные области.

Приведенный пример хорошо иллюстрирует главную причину нестабильности маршрутизаторов, работающих по протоколу RIP. Эта причина коренится в самом принципе работы дистанционно-векторных протоколов — использовании информации, полученной из «вторых рук». Действительно, маршрутизатор R2 передает маршрутизатору R1 информацию о достижимости сети 201.36.14.0, за достоверность которой он сам не отвечает.

ПРИМЕЧАНИЕ

Не следует думать, что при любых отказах интерфейсов и маршрутизаторов в сетях возникают маршрутные петли. Если бы маршрутизатор R1 успел передать сообщение о недостижимости сети 201.36.14.0 раньше ложной информации маршрутизатора R2, то маршрутная петля не образовалась бы. Так что маршрутные петли даже без дополнительных методов борьбы с ними возникают в среднем не более чем в половине потенциально возможных случаев.

Методы борьбы с ложными маршрутами в протоколе RIP

Хотя протокол RIP не в состоянии полностью исключить в сети переходные состояния, когда некоторые маршрутизаторы пользуются устаревшей информацией о несуществующих маршрутах, имеется несколько методов, которые во многих случаях решают подобные проблемы.

Проблема с петлей, образующейся между соседними маршрутизаторами, надежно решается с помощью метода **расщепления горизонта**. Этот метод заключается в том, что маршрутная информация о некоторой сети, хранящаяся в таблице маршрутизации, никогда не передается тому маршрутизатору, от которого она получена.

Практически все сегодняшние маршрутизаторы, работающие по протоколу RIP, используют технику расщепления горизонта. Если бы маршрутизатор R2 в рассмотренном ранее примере поддерживал технику расщепления горизонта, то он бы не передал маршрутизатору R1 устаревшую информацию о сети 201.36.14.0, так как получил он ее именно от маршрутизатора R1.

Однако расщепление горизонта не помогает в тех случаях, когда петли образуются не двумя, а большим числом маршрутизаторов. Рассмотрим более детально ситуацию, которая возникнет в сети, приведенной на рис. 17.17, в случае потери связи маршрутизатора R1 с сетью 201.36.14.0. Пусть все маршрутизаторы этой сети поддерживают технику расщепления горизонта. Маршрутизаторы R2 и R3 не будут возвращать маршрутизатору в этой ситуации данные о сети 201.36.14.0 с метрикой 2, так как они получили эту информацию от маршрутизатора R1. Однако они будут передавать маршрутизатору информацию о достижимости сети 201.36.14.0 с метрикой 4 через себя, так как получили эту информацию по сложному маршруту, а не непосредственно от маршрутизатора R1. Например, маршрутизатор R2 получает эту информацию по цепочке R4-R3-R1, поэтому маршрутизатор R1 снова может быть обманут, пока каждый из маршрутизаторов в цепочке R3-R4-R2 не вычеркнет запись о достижимости сети 201.36.14.0.

Для предотвращения зацикливания пакетов по составным петлям при отказах связей применяются два других приема, называемые триггерными обновлениями и замораживанием изменений.

Прием **триггерных обновлений** состоит в том, что маршрутизатор, получив данные об изменении метрики до какой-либо сети, не ждет истечения периода передачи таблицы маршрутизации, а передает данные об изменившемся маршруте немедленно. Этот прием может во многих случаях предотвратить передачу устаревших сведений об отказавшем маршруте, но он перегружает сеть служебными сообщениями, поэтому триггерные объявления также делаются с некоторой задержкой. По этой причине возможна ситуация, когда регулярное обновление в каком-либо маршрутизаторе чуть опережает по времени приход триггерного обновления от предыдущего в цепочке маршрутизатора, и данный маршрутизатор успевает передать по сети устаревшую информацию о несуществующем маршруте.

Второй прием — **замораживание изменений** — позволяет исключить подобные ситуации. Он связан с введением тайм-аута на принятие новых данных о сети, которая только что стала недоступной. Этот тайм-аут предотвращает принятие устаревших сведений о некотором маршруте от тех маршрутизаторов, которые находятся на некотором расстоянии от отказавшей связи и передают устаревшие сведения о ее работоспособности. Предполагается, что в течение тайм-аута «замораживания изменений» эти маршрутизаторы вычеркнут данный маршрут из своих таблиц, так как не получат о нем новых записей и не будут распространять устаревшие сведения по сети.

Протокол OSPF

Протокол OSPF (Open Shortest Path First — выбор кратчайшего пути первым) является последним (он принят в 1991 году) протоколом, основанном на алгоритме состояния связей, и обладает многими особенностями, ориентированными на применение в больших гетерогенных сетях.

Два этапа построения таблицы маршрутизации

OSPF разбивает процедуру построения таблицы маршрутизации на два этапа, к первому относится построение и поддержание базы данных о состоянии связей сети, ко второму — нахождение оптимальных маршрутов и генерация таблицы маршрутизации.

Построение и поддержание базы данных о состоянии связей сети. Связи сети могут быть представлены в виде графа, в котором вершинами графа являются маршрутизаторы и подсети, а ребрами — связи между ними (рис. 17.18). Каждый маршрутизатор обменивается со своими соседями той информацией о графе сети, которой он располагает к данному моменту Этот процесс похож на процесс распространения векторов расстояний до сетей в протоколе RIP, однако сама информация качественно иная — это информация о *топологии* сети. Сообщения, с помощью которых распространяется топологическая информация, называются **объявлениями о состоянии связей** (Link State Advertisement, LSA) сети. При транзитной передаче объявлений LSA маршрутизаторы не модифицируют информацию, как это происходит в дистанционно-векторных протоколах, в частности в RIP, а передают ее в неизменном виде. В результате все маршрутизаторы сети сохраняют в своей памяти идентичные сведения о текущей конфигурации графа связей сети.

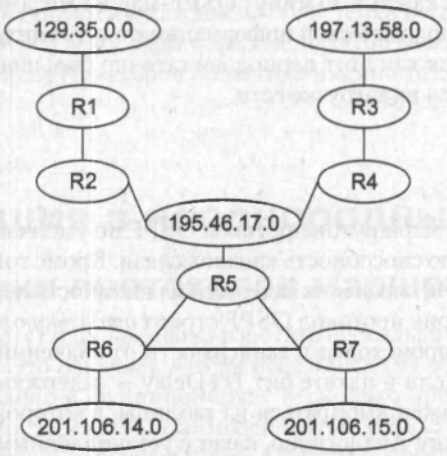


Рис. 17.18. Граф сети, построенный протоколом OSPF

Для контроля состояния связей и соседних маршрутизаторов OSPF-маршрутизаторы передают друг другу особые сообщения HELLO каждые 10 секунд . Небольшой объем этих сообщений делает возможным частое тестирование состояния соседей и связей с ними. В том случае, когда сообщения HELLO перестают поступать от какого-либо непосредственного соседа, маршрутизатор делает вывод о том, что состояние связи изменилось с работоспособного на неработоспособное и вносит соответствующие коррективы в свою топологическую базу данных. Одновременно он отсылает всем непосредственным соседям объявление LSA об этом изменении, те также вносят исправления в свои базы данных и, в свою очередь, рассылают данное объявление LSA своим непосредственным соседям.

Нахождение оптимальных маршрутов и генерация таблицы маршрутизации. Задача нахождения оптимального пути на графе является достаточно сложной и трудоемкой. В протоколе OSPF для ее решения используется итеративный алгоритм Дийкстры. Каждый

маршрутизатор сети, действуя в соответствии с этим алгоритмом, ищет оптимальные маршруты от своих интерфейсов до всех известных ему подсетей. В каждом найденном таким образом маршруте запоминается только один шаг — до следующего маршрутизатора. Данные об этом шаге и попадают в таблицу маршрутизации.

Если состояние связей в сети изменилось и произошла корректировка графа сети, каждый маршрутизатор заново ищет оптимальные маршруты и корректирует свою таблицу маршрутизации. Аналогичный процесс происходит и в том случае, когда в сети появляется новая связь или новый сосед, объявляющий о себе с помощью своих сообщений HELLO. При работе протокола OSPF конвергенция таблиц маршрутизации к новому согласованному состоянию происходит достаточно быстро, быстрее, чем в сетях, в которых работают дистанционно-векторные протоколы. Это время состоит из времени распространения по сети объявления LSA и времени работы алгоритма Дийкстры, который обладает быстрой сходимостью. Однако вычислительная сложность этого алгоритма предъявляет высокие требования к мощности процессора маршрутизатора.

Когда состояние сети не меняется, то объявления о связях не генерируются, топологические базы данных и таблицы маршрутизации не корректируются, что экономит пропускную способность сети и вычислительные ресурсы маршрутизаторов. Однако у этого правила есть исключение: каждые 30 минут OSPF-маршрутизаторы обмениваются всеми записями базы данных топологической информации, то есть синхронизируют их для более надежной работы сети. Так как этот период достаточно большой, то данное исключение незначительно сказывается на загрузке сети.

Метрики

При поиске оптимальных маршрутов протокол OSPF по умолчанию использует метрику, учитывающую пропускную способность каналов связи. Кроме того, допускается применение двух других метрик, учитывающих задержки и надежность передачи пакетов каналами связи. Для каждой из метрик протокол OSPF строит *отдельную* таблицу маршрутизации. Выбор нужной таблицы происходит в зависимости от значений битов TOS в заголовке пришедшего IP-пакета. Если в пакете бит D (Delay — задержка) установлен в 1, то для этого пакета маршрут должен выбираться из таблицы, в которой содержатся маршруты, минимизирующие задержку. Аналогично, пакет с установленным битом T (Throughput – пропускная способность) должен маршрутизироваться по таблице, построенной с учетом пропускной способности каналов, а установленный в единицу бит R (Reliability — надежность) указывает на то, что должна использоваться таблица, для построения которой критерием оптимизации служит надежность доставки.

Протокол OSPF поддерживает стандартные для многих протоколов (например, для протокола покрывающего дерева) значения расстояний для метрики, отражающей пропускную способность: так, для сети Ethernet она равна 10, для Fast Ethernet — 1, для канала T-1[1], обладающего пропускной способностью 1,544 Мбит/с, — 65, для канала с пропускной способностью 56 Кбит/с — 1785. При наличии высокоскоростных каналов, таких как Gigabit Ethernet или STM-16/64, администратору нужно задать другую шкалу скоростей, назначив единичное расстояние наиболее скоростному каналу.

[1] T-1 — это цифровой канал технологии PDH, рассматривавшейся в главе 11.

При выборе оптимального пути на графе с каждым ребром графа связывается метрика, которая добавляется к пути, если данное ребро в него входит. Пусть в приведенном на рис. 17.18 примере маршрутизатор R5 связан с маршрутизаторами R6 и R7 каналами T-1, а маршрутизаторы R6 и R7 связаны между собой каналом 56 Кбит/с. Тогда R7 определит оптимальный маршрут до сети 201.106.14.0 как составной, проходящий сначала через R5, а затем через R6, поскольку у этого маршрута метрика будет равна 65 + 65 = 130 единиц. Непосредственный маршрут через R6 не будет оптимальным, так как его метрика равна 1785.

Протокол OSPF разрешает хранить в таблице маршрутизации несколько маршрутов к одной сети, если они обладают равными метриками. В таких случаях маршрутизатор может работать в режиме баланса загрузки маршрутов, отправляя пакеты попеременно по каждому из маршрутов.

К сожалению, вычислительная сложность протокола OSPF быстро растет с увеличением размера сети. Для преодоления этого недостатка в протоколе OSPF вводится понятие **области сети**. Маршрутизаторы, принадлежащие некоторой области, строят граф связей только для этой области, что упрощает задачу. Между областями информация о связях не передается, а пограничные для областей маршрутизаторы обмениваются только информацией об адресах сетей, имеющихся в каждой из областей, и *расстоянием от пограничного маршрутизатора до каждой сети*. При передаче пакетов между областями выбирается один из пограничных маршрутизаторов области, а именно тот, у которого расстояние до нужной сети меньше.

Маршрутизация в неоднородных сетях

Взаимодействие протоколов маршрутизации

В одной и той же сети могут одновременно работать несколько разных протоколов маршрутизации (рис. 17.19). Это означает, что на некоторых (не обязательно всех) маршрутизаторах сети установлено и функционирует несколько протоколов маршрутизации, но при этом, естественно, через сеть взаимодействуют только одноименные протоколы. То есть если маршрутизатор 1 поддерживает, например, протоколы RIP и OSPF, маршрутизатор 2 — только RIP, а маршрутизатор 3 — только OSPF, то маршрутизатор 1 будет взаимодействовать с маршрутизатором 2 по протоколу RIP, с маршрутизатором 3 — по OSPF, а маршрутизаторы 2 и 3 вообще непосредственно друг с другом взаимодействовать не смогут.

В маршрутизаторе, который поддерживает одновременно несколько протоколов, каждая запись в таблице является результатом работы одного из этих протоколов. Если информация о некоторой сети появляется от нескольких протоколов, то для однозначности выбора маршрута (а данные разных протоколов могут вести к разным рациональным маршрутам) устанавливаются *приоритеты протоколов маршрутизации*. Обычно предпочтение отдается протоколам LSA, как располагающим более полной информацией о сети по сравнению с протоколами DVA. В некоторых ОС в формах вывода на экран и печать в каждой записи таблицы маршрутизации имеется отметка о том, с помощью какого протокола маршрутизации эта запись получена. Но даже если эта отметка на экран и не выводится, она обязательно имеется во внутреннем представлении таблицы маршрутизации.

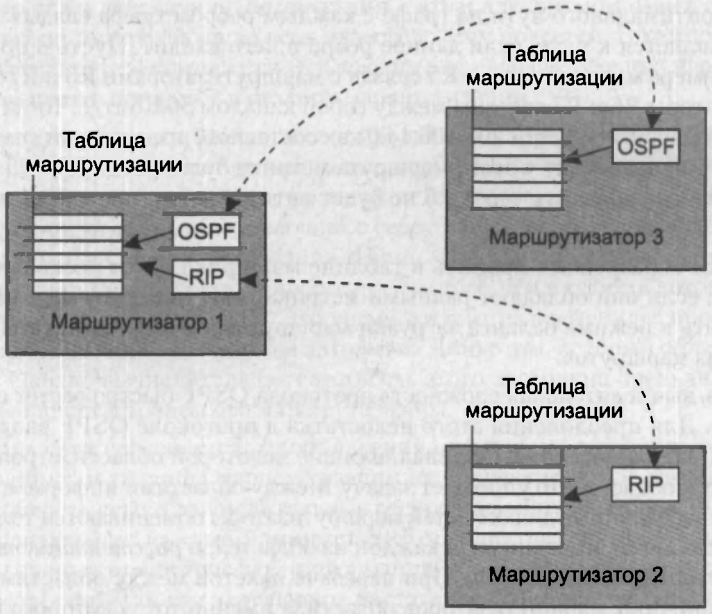


Рис. 17.19. Применение нескольких протоколов маршрутизации в одной сети

По умолчанию каждый протокол маршрутизации, работающий на определенном маршрутизаторе, распространяет только «собственную» информацию, то есть ту информацию, которая была получена данным маршрутизатором по данному протоколу. Например, если о маршруте к некоторой сети маршрутизатор узнал по протоколу RIP, то и распространять по сети объявления об этом маршруте он будет с помощью протокола RIP.

Однако такой «избирательный» режим работы маршрутизаторов ставит невидимые барьеры на пути распространения маршрутной информации, создавая в составной сети области взаимной недостижимости. Задача маршрутизации решалась бы эффективнее, если бы маршрутизаторы могли обмениваться маршрутной информацией, полученной разными протоколами маршрутизации. Такая возможность реализуется в особом режиме работы маршрутизатора, называемом *перераспределением*. Этот режим позволяет одному протоколу маршрутизации использовать не только «свои», но и «чужие» записи таблицы маршрутизации, полученные с помощью другого протокола маршрутизации, указанного при конфигурировании.

Как видим, применение нескольких протоколов маршрутизации даже в пределах небольшой составной сети — дело не простое, от администратора требуется провести определенную работу по конфигурированию каждого маршрутизатора. Очевидно, что для крупных составных сетей нужно качественно иное решение.

Внутренние и внешние шлюзовые протоколы

Такое решение было найдено для самой крупной на сегодня составной сети — Интернета. Это решение базируется на понятии автономной системы.

Автономная система (Autonomous System, AS) — это совокупность сетей под единым административным управлением, обеспечивающим общую для всех входящих в автономную систему маршрутизаторов политику маршрутизации.

Обычно автономной системой управляет один поставщик услуг Интернета, самостоятельно выбирая, какие протоколы маршрутизации должны использоваться в некоторой автономной системе и каким образом между ними должно выполняться перераспределение маршрутной информации. Крупные поставщики услуг и корпорации могут представить свою составную сеть как набор нескольких автономных систем. Регистрация автономных систем происходит централизованно, как и регистрация IP-адресов и DNS-имен. Номер автономной системы состоит из 16 разрядов и никак не связан с префиксами IP-адресов входящих в нее сетей.

В соответствии с этой концепцией Интернет выглядит как набор взаимосвязанных автономных систем, каждая из которых состоит из взаимосвязанных сетей (рис. 17.20), соединенными **внешними шлюзами**.

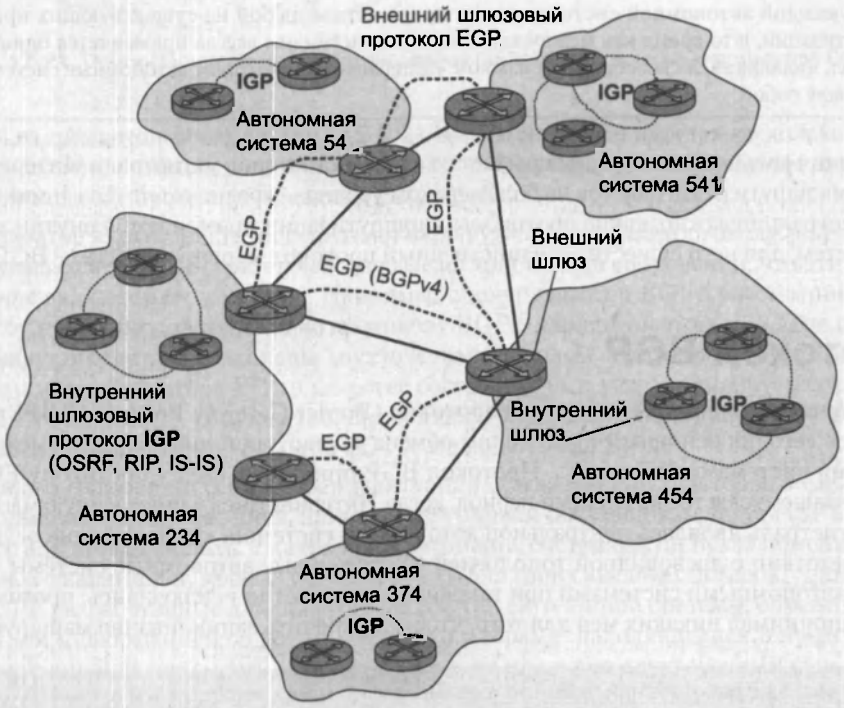


Рис. 17.20. Автономные системы Интернета

Основная цель деления Интернета на автономные системы — обеспечение многоуровневого подхода к маршрутизации. До введения автономных систем предполагался двухуровневый подход, то есть сначала маршрут определялся как *последовательность сетей*, а затем вел непосредственно к заданному узлу в конечной сети (именно этот подход мы использовали до сих пор).

С появлением автономных систем появляется третий, верхний, уровень маршрутизации — теперь сначала маршрут определяется как *последовательность автономных систем*, затем — как *последовательность сетей* и только потом ведет к конечному узлу.

Выбор маршрута между автономными системами осуществляют внешние шлюзы, использующие особый тип протокола маршрутизации, так называемый **внешний шлюзовой протокол** (Exterior Gateway Protocol, EGP). В настоящее время для работы в такой роли сообщество Интернета утвердило стандартный **пограничный шлюзовой протокол** версии 4 (Border Gateway Protocol, BGPv4). В качестве адреса следующего маршрутизатора в протоколе BGPv4 указывается адрес точки входа в соседнюю автономную систему

За *маршрут внутри автономной системы* отвечают **внутренние шлюзовые протоколы** (Interior Gateway Protocol, IGP). К числу IGP относятся знакомые нам протоколы RIP, OSPF и IS-IS. В случае транзитной автономной системы эти протоколы указывают точную последовательность маршрутизаторов от точки входа в автономную систему до точки выхода из нее.

ПРИМЕЧАНИЕ

Внутри каждой автономной системы может применяться любой из существующих протоколов маршрутизации, в то время как между автономными системами всегда применяется один и тот же протокол, являющийся своеобразным языком «эсперанто», на котором автономные системы общаются между собой.

Концепция автономных систем скрывает от администраторов магистрали Интернета проблемы маршрутизации пакетов на более низком уровне — уровне сетей. Для администратора магистрали неважно, какие протоколы маршрутизации применяются внутри автономных систем, для него существует единственный протокол маршрутизации — BGPv4.

Протокол BGP

Пограничный (внешний) шлюзовой протокол (Border Gateway Protocol, BGP) версии 4 является сегодня основным протоколом обмена маршрутной информацией между автономными системами Интернета. Протокол BGP пришел на смену протоколу EGP[1], использовавшемуся в тот начальный период, когда Интернет имел единственную магистраль. Эта магистраль являлась центральной автономной системой, к которой присоединялись в соответствии с древовидной топологией все остальные автономные системы. Так как между автономными системами при такой структуре петли исключались, протокол EGP не предпринимал никаких мер для того, чтобы исключить зацикливание маршрутов.

BGPv4 успешно работает при любой топологии связей между автономными системами, что соответствует современному состоянию Интернета.

Поясним основные принципы работы BGP на примере (рис. 17.21).

[1] EGP в данном случае является названием конкретного протокола маршрутизации. Напомним, что аббревиатура EGP служит также названием класса внешних шлюзовых протоколов, используемых для маршрутизации между автономными системами, что вносит некоторую путаницу.

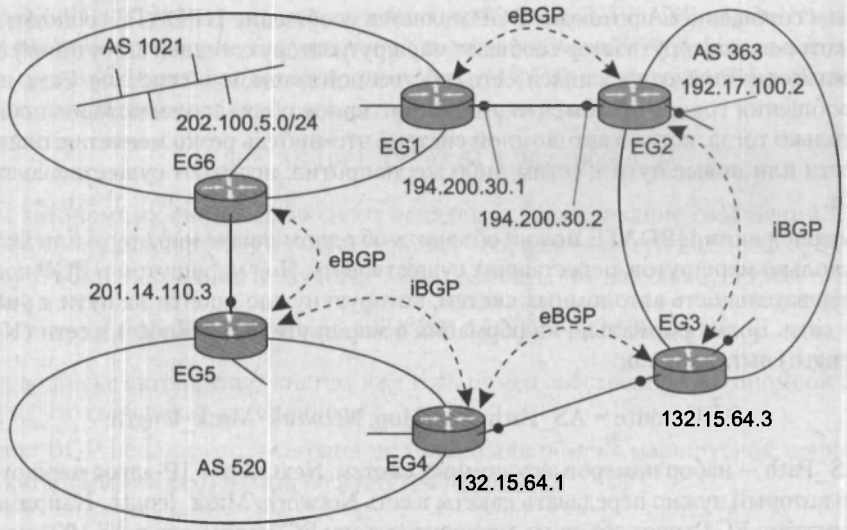


Рис. 17.21. Поиск маршрута между автономными системами с помощью протокола BGP

В каждой из трех автономных систем (AS 1021, AS 363 и AS 520) имеется несколько маршрутизаторов, исполняющих роль внешних шлюзов. На каждом из них работает протокол BGP, с помощью которого они общаются между собой.

Маршрутизатор взаимодействует с другими маршрутизаторами по протоколу BGP только в том случае, если администратор *явно* указывает при конфигурировании, что эти маршрутизаторы являются его *соседями*. Например, маршрутизатор EG1 в рассматриваемом примере будет взаимодействовать по протоколу BGP с маршрутизатором EG2 не потому, что эти маршрутизаторы соединены двухточечным каналом, а потому, что при конфигурировании маршрутизатора EG1 в качестве соседа ему был указан маршрутизатор EG2 (с адресом 194.200.30.2). Аналогично, при конфигурировании маршрутизатора EG2 его соседом был назначен маршрутизатор EG1 (с адресом 194.200.30.1).

Такой способ взаимодействия удобен в ситуации, когда маршрутизаторы, обменивающиеся маршрутной информацией, принадлежат разным поставщикам услуг (ISP). Администратор ISP может решать, с какими автономными системами он будет обмениваться трафиком, а с какими нет, задавая список соседей для своих внешних шлюзов. Протоколы RIP и OSPF, разработанные для применения внутри автономной системы, обмениваются маршрутной информацией со всеми маршрутизаторами, находящимися в пределах их непосредственной досягаемости (по локальной сети или через двухточечный канал). Это означает, что информация обо всех сетях появляется в таблице маршрутизации каждого маршрутизатора, так что каждая сеть оказывается достижимой для каждой. В корпоративной сети это нормальная ситуация, а в сетях ISP нет, поэтому протокол BGP и исполняет здесь особую роль.

Для установления сеанса с указанными соседями BGP-маршрутизаторы используют протокол TCP (порт 179). При установлении BGP-сеанса могут применяться разнообразные способы аутентификации маршрутизаторов, повышающие безопасность работы автономных систем.

Основным сообщением протокола BGP является сообщение UPDATE (обновить), с помощью которого маршрутизатор сообщает маршрутизатору соседней автономной системы о достижимости сетей, относящихся к его собственной автономной системе. Само название этого сообщения говорит о том, что это — триггерное объявление, которое посылается соседу только тогда, когда в автономной системе что-нибудь резко меняется: появляются новые сети или новые пути к сетям либо же, напротив, исчезают существовавшие сети или пути.

В одном сообщении UPDATE можно объявить об одном новом маршруте или аннулировать несколько маршрутов, переставших существовать. Под маршрутом в BGP понимается последовательность автономных систем, которую нужно пройти на пути к указанной в адресе сети. Более формально информация о маршруте (BGP Route) к сети (Network/Mask_length) выглядит так:

BGP Route = AS_Path; NextHop; Network/Mask_length;

Здесь AS_Path — набор номеров автономных систем, NextHop — IP-адрес маршрутизатора, через который нужно передавать пакеты в сеть Network/Mask_length. Например, если маршрутизатор EG1 хочет объявить маршрутизатору EG2 о том, что в AS 1021 появилась новая сеть 202.100.5.0/24, то он формирует такое сообщение:

AS 1021; 194.200.30.1; 202.100.5.0/24,

после чего передает его маршрутизатору EG2 автономной системы AS 363 (с которым у него, конечно, должен быть установлен BGP-сеанс).

Маршрутизатор EG2, получив сообщение UPDATE, запоминает в своей таблице маршрутизации информацию о сети 202.100.5.0/24 вместе с адресом следующего маршрутизатора 194.200.30.1 и отметкой о том, что эта информация была получена по протоколу BGP. Маршрутизатор EG2 обменивается маршрутной информацией с внутренними шлюзами системы AS 363 по какому-либо протоколу группы IGP, например OSPF. Если у EG2 установлен режим перераспределения маршрутов BGP в маршруты OSPF, то все внутренние шлюзы AS 363 узнают о существовании сети 202.100.5.0/24 с помощью объявления OSPF, которое будет внешним. В качестве адреса следующего маршрутизатора маршрутизатор EG2 начнет теперь объявлять адрес собственного внутреннего интерфейса, например 192.17.100.2.

Однако для распространения сообщения о сети 202.100.5.0/24 в другие автономные системы, например в AS 520, протокол OSPF использоваться не может. Маршрутизатор EG3, связанный с маршрутизатором EG4 автономной системы 520, должен пользоваться протоколом BGP, генерируя сообщение UPDATE нужного формата. Для решения этой задачи он не может задействовать информацию о сети 202.100.5.0/24, полученную от протокола OSPF через один из своих внутренних интерфейсов, так как она имеет другой формат и не содержит, например, сведений о номере автономной системы, в которой находится эта сеть.

Проблема решается за счет того, что маршрутизаторы EG2 и EG3 также устанавливают между собой BGP-сеанс, хотя они и принадлежат одной и той же автономной системе. Такая реализация протокола BGP называется **внутренней** (Interior BGP, iBGP), в отличие от основной, **внешней** (Exterior BGP, eBGP). В результате маршрутизатор EG3 получает нужную информацию от маршрутизатора EG2 и передает ее внешнему соседу — марш-

рутизатору EG4. При формировании нового сообщения UPDATE маршрутизатор EG3 трансформирует сообщение, полученное от маршрутизатора EG2, добавляя в список автономных систем собственную автономную систему AS 520, а полученный адрес следующего маршрутизатора заменяет адресом собственного интерфейса:

AS 363, AS 1021; 132.15.64.3; 202.100.5.0/24.

Номера автономных систем позволяют исключать зацикливание сообщений UPDATE. Например, когда маршрутизатор EG5 передаст сообщение о сети 202.100.5.0/24 маршрутизатору EG6, то последний не будет его использовать, так как оно будет иметь вид:

AS 520, AS 363, AS 1021; 201.14.110.3; 202.100.5.0/24.

Так как в списке автономных систем уже есть номер собственной автономной системы, очевидно, что сообщение зациклилось.

Протокол BGP используется сегодня не только для обмена маршрутной информацией между автономными системами, но и внутри них.

Протокол ICMP

Протокол межсетевых управляющих сообщений (Internet Control Message Protocol, ICMP) (RFC 792) является вспомогательным протоколом, использующимся для диагностики и мониторинга сети.

Можно представить ряд ситуаций, когда протокол IP не может доставить пакет адресату, например истекает время жизни пакета, в таблице маршрутизации отсутствует маршрут к заданному в пакете адресу назначения, пакет не проходит проверку по контрольной сумме, шлюз не имеет достаточно места в своем буфере для передачи какого-либо пакета и т. д., и т. п.

Как мы не раз отмечали, протокол IP доставляет данные, руководствуясь принципом «по возможности», то есть не предпринимает мер для гарантированной передачи данных адресату. Это свойство «необязательности» протокола IP компенсируется протоколами более высоких уровней стека TCP/IP, например TCP на транспортном уровне и в какой-то степени DNS на прикладном уровне. Они берут на себя обязанности по обеспечению надежности, применяя такие известные приемы, как нумерация сообщений, подтверждение доставки, повторная посылка данных.

Протокол ICMP также служит дополнением, компенсирующим ненадежность протокола IP, но несколько *другого рода*. Он не предназначен для исправления возникших при передаче пакета проблем: если пакет потерян, ICMP не может послать его заново. Задача ICMP другая — он является *средством оповещения* отправителя о «несчастных случаях», произошедших с его пакетами. Пусть, например, протокол IP, работающий на каком-либо маршрутизаторе, обнаружил, что пакет для дальнейшей передачи по маршруту необходимо фрагментировать, но в пакете установлен признак DF (не фрагментировать). Протокол IP, обнаруживший, что он не может передать IP-пакет далее по сети, прежде чем отбросить пакет, должен отправить *диагностическое* ICMP-сообщение конечному узлу-источнику.

Для передачи по сети ICMP-сообщение инкапсулируется в поле данных IP-пакета. IP-адрес узла-источника определяется из заголовка пакета, вызвавшего инцидент.

Сообщение, прибывшее в узел-источник, может быть обработано там либо ядром операционной системы, либо протоколами транспортного и прикладного уровней, либо приложениями, либо просто проигнорированы. Важно, что обработка ICMP-сообщений не входит в обязанности протоколов IP и ICMP.

Заметим, что некоторые из пакетов могут исчезнуть в сети, не вызвав при этом никаких оповещений. В частности, протокол ICMP не предусматривает передачу сообщений о проблемах, возникающих при обработке IP-пакетов, несущих ICMP-сообщения об ошибках. Такое решение было принято разработчиками протокола, чтобы не порождать «штормы» в сетях, когда количество сообщений об ошибках лавинообразно возрастает.

Особенностью протокола ICMP является функциональное разнообразие решаемых задач, а следовательно, и связанных с этим сообщений. Все типы сообщений имеют один и тот же формат (рис. 17.22), однако интерпретация полей существенно зависит от того, к какому типу относится сообщение.

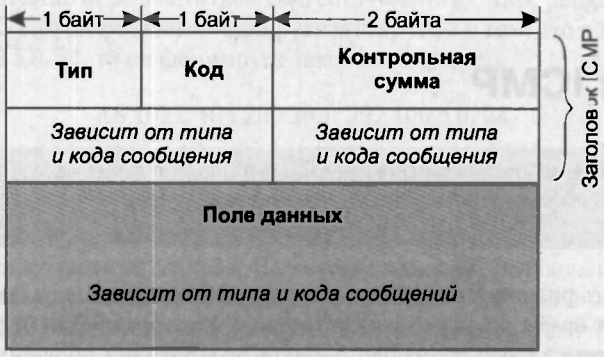


Рис. 17.22. Формат ICMP-сообщения

Заголовок ICMP-сообщения состоит из 8 байт:

- **тип** (1 байт) — числовой идентификатор типа сообщения;
- **код** (1 байт) — числовой идентификатор, более тонко дифференцирующий тип ошибки;
- **контрольная сумма** (2 байта) — подсчитывается для всего ICMP-сообщения.

Содержимое оставшихся четырех байтов в заголовке и поле данных зависит от значений полей типа и кода.

На рис. 17.23 показана таблица основных типов ICMP-сообщений. Эти сообщения можно разделить на две группы (помеченные на рисунке условными символами):

- сообщения об *ошибках*;
- сообщения *запрос-ответ*.

Сообщения типа запрос-ответ связаны в пары: эхо-запрос — эхо-ответ, запрос маски — ответ маски, запрос времени — ответ времени. Отправитель сообщения-запроса всегда рассчитывает на получение соответствующего сообщения-ответа.

Таблица типов ICMP-сообщений

Значение в поле «Тип»	Тип сообщения
0	Эхо-ответ
3	Узел назначения недостижим
4	Подавление источника
5	Перенаправление маршрута
8	Эхо-запрос
11	Истечение времени диаграммы
12	Проблема с параметрами пакета
13	Запрос отметки времени
14	Ответ отметки времени
17	Запрос маски
18	Ответ маски

Таблица кодов причин ошибок 3

Код	Причина
0	Сеть недостижима
1	Узел недостижим
2	Протокол недостижим
3	Порт недостижим
4	Ошибка фрагментации
5	Ошибка в маршруте источника
6	Сеть назначения не известна
7	Узел назначения не известен
8	Узел-источник изолирован
9	Административный запрет
	. . . . .

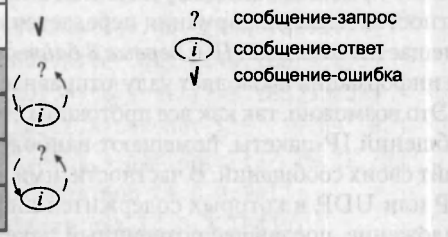


Рис. 17.23. Типы и коды ICMP-сообщений

Сообщения, относящиеся к группе сообщений об ошибках, конкретизируются уточняющим кодом. На рисунке показан фрагмент таблицы кодов для сообщения об ошибке недостижимости узла назначения, имеющей тип 3. Из таблицы мы видим, что это сообщение может быть вызвано различными причинами, такими как неверный адрес сети или узла (код 0 или 1), отсутствием на конечном узле-адресате необходимого протокола прикладного уровня (код 2 — «протокол недостижим») или открытого порта UDP/TCP (код 3 — «порт недостижим»). Узел (или сеть) назначения может быть также недостижим по причине временной неработоспособности аппаратуры или из-за того, что маршрутизатор не имеет данных о пути к сети назначения. Всего таблица содержит 15 кодов. Аналогичные таблицы кодов существуют и для других типов сообщений об ошибках.

Утилита traceroute

В качестве примера рассмотрим использование сообщений об ошибках в популярной утилите мониторинга сети traceroute.

Когда маршрутизатор не может передать или доставить IP-пакет, он отсылает узлу, отправившему этот пакет, сообщение о недостижимости узла назначения. Формат этого сообщения показан на рис. 17.24. В поле типа помещается значение 3, а в поле кода — значение из диапазона 0–15, уточняющее причину, по которой пакет не был доставлен. Следующие за полем контрольной суммы четыре байта заголовка *не используются* и заполняются нулями.

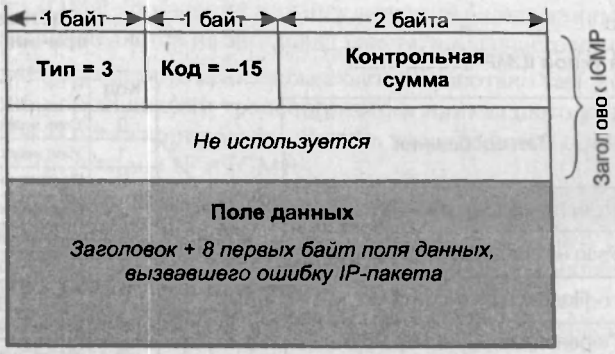


Рис. 17.24. Формат ICMP-сообщения об ошибке недостижимости узла назначения

Помимо причины ошибки, указанной в заголовке (в полях типа и кода), дополнительная диагностическая информация передается в поле данных ICMP-сообщения. Именно туда помещается *заголовок IP и первые 8 байт данных* того IP-пакета, который вызвал ошибку. Эта информация позволяет узлу-отправителю еще точнее диагностировать причину ошибки. Это возможно, так как все протоколы стека TCP/IP, использующие для передачи своих сообщений IP-пакеты, помещают наиболее важную для анализа информацию в первые 8 байт своих сообщений. В частности, ими вполне могут оказаться первые 8 байт заголовка TCP или UDP, в которых содержится информация (номер порта), идентифицирующая приложение, пославшее потерянный пакет. Следовательно, при разработке приложения можно предусмотреть встроенные средства реакции на сообщения о недоставленных пакетах.

ICMP-сообщения об ошибках лежат в основе работы популярной утилиты traceroute для Unix, имеющей в Windows название tracert. Эта утилита позволяет проследить маршрут до удаленного хоста, определить среднее время оборота (RTT), IP-адрес и в некоторых случаях доменное имя каждого промежуточного маршрутизатора. Такая информация помогает найти маршрутизатор, на котором обрывается путь пакета к удаленному хосту.

Утилита traceroute осуществляет трассировку маршрута, посылая серию обычных IP-пакетов в конечную точку изучаемого маршрута. Идея метода состоит в следующем. Значение времени жизни (TTL) первого отправляемого пакета устанавливается равным 1. Когда протокол IP первого маршрутизатора принимает этот пакет, то он в соответствии со своим алгоритмом уменьшает значение TTL на 1 и получает 0. Маршрутизатор отбрасывает пакет с нулевым временем жизни и возвращает узлу-источнику ICMP-сообщение об ошибке истечения времени дейтаграммы (значение поля типа равно 11) вместе с заголовком IP и первыми 8 байтами потерянного пакета.

Получив ICMP-сообщение о причине недоставки пакета, утилита traceroute запоминает адрес первого маршрутизатора (который извлекает из заголовка IP-пакета, несущего ICMP-сообщение).

Затем traceroute посылает следующий IP-пакет, но теперь со значением TTL, равным 2. Этот пакет благополучно проходит первый маршрутизатор, но «умирает» на втором, о чем немедленно отправляется аналогичное ICMP-сообщение об ошибке истечения времени дейтаграммы. Утилита traceroute запоминает адрес второго маршрутизатора и т. д. Такие

действия выполняются с каждым маршрутизатором вдоль маршрута вплоть до узла назначения или неисправного маршрутизатора. Мы рассматриваем работу утилиты traceroute весьма схематично, но и этого достаточно, чтобы оценить изящество идеи, лежащей в основе ее работы. Остальные ICMP-сообщения об ошибках имеют такой же формат и отличаются друг от друга только значениями полей типа и кода.

Далее приведена копия экранной формы, выведенной утилитой tracert (Windows) при трассировке хоста ds.internic.net [198.49.45.29]:

```
 1 311 ms 290 ms 261 ms 144.206.192.100
 2 281 ms 300 ms 271 ms 194.85.73.5
 3 2023 ms 290 ms 311 ms moscow-m9-2-S5.relcom.eu.net [193.124.254.37]
 4 290 ms 261 ms 280 ms MSK-M9-13.Relcom.EU.net [193.125.15.13]
 5 270 ms 281 ms 290 ms MSK.RAIL-1-ATM0-155Mb.Relcom.EU.net [193.124.254.82]
 6 300 ms 311 ms 290 ms SPB-RASCOM-1-E3-1-34Mb.Relcom.EU.net [193.124.254.78]
 7 311 ms 300 ms 300 ms Hssi11-0.GW1.STK2.ALTER.NET [146.188.33.125]
 8 311 ms 330 ms 291 ms 421.ATM6-0-0.CR2.STK2.Alter.Net [146.188.5.73]
 9 360 ms 331 ms 330 ms 219.Hssi4-0.CR2.LND1.Alter.Net [146.188.2.213]
10 351 ms 330 ms 331 ms 412.Atm5-0.BR1.LND1.Alter.net [146.188.3.205]
11 420 ms 461 ms 420 ms 167.ATM8-0-0.CR1.ATL1.Alter.Net [137.39.69.182]
12 461 ms 441 ms 440 ms 311.ATM12-0-0.BR1.ATL1.Alter.Net [137.39.21.73]
13 451 ms 410 ms 431 ms atlanta1-br1.bbnplanet.net [4.0.2.141]
14 420 ms 411 ms 410 ms vienna1-br2.bbnplanet.net [4.0.3.154]
15 411 ms 430 ms 2514 ms vienna1-nbr3.bbnplanet.net [4.0.3.150]
16 430 ms 421 ms 441 ms vienna1-nbr2.bbnplanet.net [4.0.5.45]
17 431 ms 451 ms 420 ms cambridge1-br1.bbnplanet.net [4.0.5.42]
18 450 ms 461 ms 441 M C cambridge1-cr14.bbnplanet.net [4.0.3.94]
19 451 M C 461 M C 460 M C attbcstoll.bbnplanet.net [206.34.99.38]
20 501 M C 460 M C 481 M C shutdown.ds.internic.net [198.49.45.29]
```

Последовательность строк соответствует последовательности маршрутизаторов, образующих маршрут к заданному узлу Первое число в строке — число хопов до соответствующего маршрутизатора. Утилита traceroute тестирует каждый маршрутизатор трижды, поэтому следующие три числа в строке — это значения RTT, вычисленные путем посылки трех пакетов, время жизни которых истекло на этом маршрутизаторе. Если ответ от какого-либо маршрутизатора не приходит за заданное время, то вместо времени на экране печатается звездочка (*).

Далее идут IP-адрес и доменное имя (если оно имеется) маршрутизатора. Видно, что почти все интерфейсы маршрутизаторов поставщиков услуг Интернета зарегистрированы в службе DNS, а первые два, относящиеся к локальным маршрутизаторам, — нет.

Еще раз подчеркнем, что время, указанное в каждой строке, это не время прохождения пакетов между двумя соседними маршрутизаторами, а время, за которое пакет проделывает путь от источника до соответствующего маршрутизатора и обратно. Так как ситуация в Интернете с загрузкой маршрутизаторов постоянно меняется, то время достижимости маршрутизаторов не всегда нарастает монотонно, а может изменяться достаточно произвольным образом.

Утилита ping

А сейчас давайте рассмотрим представителей другой группы ICMP-сообщений — **эхо-запросы** и **эхо-ответы** и поговорим об использовании этих сообщений в известной утилите ping.

Эхо-запрос и эхо-ответ, в совокупности называемые **эхо-протоколом**, представляют собой очень простое средство мониторинга сети. Компьютер или маршрутизатор посылает по составной сети ICMP-сообщение эхо-запроса, указывая в нем IP-адрес узла, достижимость которого нужно проверить. Узел, получивший эхо-запрос, формирует и отправляет эхо-ответ отправителю запроса. Так как эхо-запрос и эхо-ответ передаются по сети внутри IP-пакетов, то их успешная доставка означает нормальное функционирование всей транспортной системы составной сети.

Формат эхо-запроса и эхо-ответа показан на рис. 17.25. Поле типа для эхо-ответа равно 0, для эхо-запроса — 8; поле кода всегда равно 0 и для запроса, и для ответа. В байтах 5 и 6 заголовка содержится **идентификатор запроса**, в байтах 7 и 8 — **порядковый номер**. В поле данных эхо-запроса может быть помещена произвольная информация, которая в соответствии с данным протоколом должна быть скопирована в поле данных эхо-ответа.

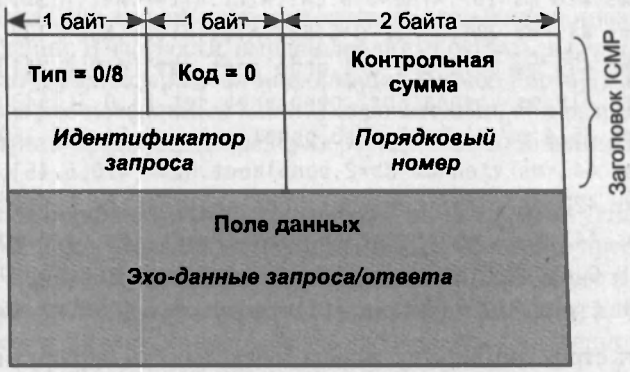


Рис. 17.25. Формат ICMP-сообщений типа эхо-запрос и эхо-ответ

Поля идентификатора запроса и порядкового номера используются одинаковым образом всеми сообщениями типа запрос-ответ. Посылая запрос, приложение помещает в эти два поля информацию, которая предназначена для последующего встраивания ее в соответствующий ответ. Сообщение-ответ копирует значения этих полей в свои поля того же назначения. Когда ответ возвращается в пункт отправки сообщения-запроса, то на основании идентификатора он может «найти и опознать» приложение, пославшее запрос. А порядковый номер используется приложением, чтобы связать полученный ответ с соответствующим запросом (учитывая, что одно приложение может выдать несколько однотипных запросов).

Утилита ping обычно посылает серию эхо-запросов к тестируемому узлу и предоставляет пользователю статистику об утерянных эхо-ответах и среднем времени реакции сети на запросы. Утилита ping выводит на экран сообщения следующего вида обо всех поступивших ответах:

```
# ping server1.citmgu.ru
Pinging server1.citmgu.ru [193.107.2.200] with 64 bytes of data:
Reply from 193.107.2.200: bytes=64 time=256ms TTL= 123
Reply from 193.107.2.200: bytes=64 time=310ms TTL= 123
Reply from 193.107.2.200: bytes=64 time=260ms TTL= 123
Reply from 193.107.2.200: bytes=64 time=146ms TTL= 123
```

Из приведенной распечатки видно, что в ответ на тестирующие запросы, посланные узлу server1.mgu.ru, было получено 4 эхо-ответа. Длина каждого сообщения составляет 64 байта. В следующей колонке помещены значения времени оборота (RTT), то есть времени от момента отправки запроса до получения ответа на этот запрос. Как видим, сеть работает достаточно нестабильно — время в последней строке отличается от времени во второй более чем в два раза. На экран выводится также оставшееся время жизни поступивших пакетов.

Выводы

В то время как задачей протокола IP является передача данных между сетевыми интерфейсами в составной сети, основная задача протоколов TCP и UDP заключается в передаче данных между прикладными процессами, выполняющимися на разных конечных узлах сети.

Протокол UDP является дейтаграммным протоколом, работающим без установления логического соединения, он не гарантирует доставку своих сообщений, а следовательно, не компенсирует ненадежность дейтаграммного протокола IP.

Системные очереди к точкам входа прикладных процессов называют портами. Порты идентифицируются номерами и однозначно определяют приложение в пределах компьютера. Если процессы представляют собой популярные общедоступные службы, такие как FTP, telnet, HTTP, TFTP, DNS и т. п., то за ними централизовано закрепляются стандартные (назначенные) номера.

TCP решает задачу надежного обмена данными путем установления логических соединений. Соединение однозначно идентифицируется парой сокетов.

Сокетом прикладного процесса называется пара из IP-адреса и номер порта.

Для управления потоком в рамках TCP-соединения используется специфический вариант алгоритма скользящего окна. Сторона-получатель передает стороне-отправителю размер окна приема в байтах.

Протоколы маршрутизации генерируют для каждого маршрутизатора согласованные таблицы маршрутизации, которые позволяют обеспечить доставку пакета по рациональному маршруту от исходной сети в сеть назначения за конечное число шагов.

Адаптивная маршрутизация обеспечивает автоматическое обновление таблиц маршрутизации после изменения конфигурации сети.

Адаптивные протоколы маршрутизации делятся на дистанционно-векторные алгоритмы (например, RIP) и алгоритмы состояния связей (например, OSPF).

Протоколы маршрутизации Интернета делятся на внешние (EGP), которые переносят маршрутную информацию между автономными системами, и внутренние (IGP), которые применяются только в пределах определенной автономной системы.

Протокол ICMP играет в сети вспомогательную роль. Он используется для диагностики и мониторинга сети. Так, в основе работы популярных утилит мониторинга IP-сетей ping и tracert лежат ICMP-сообщения.

Вопросы и задания

1. Какой объем данных получен в течение TCP-сеанса отправителем TCP-сегмента, в заголовке которого в поле квитанции помещено значение 180005? Известно, что первый полученный байт имел номер 15000.
2. Может ли работать маршрутизатор, не имея таблицы маршрутизации? Варианты ответов:
   а) может, если выполняется маршрутизация от источника;
   б) нет, это невозможно;
   в) может, если в маршрутизаторе задан маршрут по умолчанию;
   г) может, если выполняется лавинная маршрутизация.
3. Можно ли обойтись в сети без протоколов маршрутизации?
4. Система DNS может использовать для доставки своих сообщений как протокол UDP, так и TCP. Какой вариант вы считаете более предпочтительным? Аргументируйте свой ответ.
5. По какой причине в протоколе RIP расстояние в 16 хопов между сетями полагается недостижимым? Варианты ответов:
   а) поле, отведенное для хранения значения расстояния, имеет длину 4 двоичных разряда;
   б) сети, в которых работает RIP, редко бывают большими;
   в) для получения приемлемого времени сходимости алгоритма.
6. Какие параметры сети учитывают метрики, поддерживаемые протоколом OSPF? Варианты ответов:
   а) пропускная способность;
   б) количество хопов;
   в) надежность каналов связи.
7. ICMP-сообщение об ошибке не посылается, если ошибка возникла при передаче IP-пакета:
   а) несущего ICMP-сообщение об ошибке;
   б) являющегося последним фрагментом пакета;
   в) несущего ICMP-запрос;
   г) упакованного в кадр с широковещательным MAC-адресом.
8. Кому адресовано ICMP-сообщение? Варианты ответов:
   а) протоколу IP узла-отправителя пакета, вызвавшего ошибку;
   б) протоколу IP ближайшего маршрутизатора, от которого поступил пакет, вызвавший ошибку;
   в) протоколу транспортного или прикладного уровня узла-отправителя пакета, вызвавшего ошибку.
9. Предложите варианты метрики, которая одновременно учитывает пропускную способность, надежность и задержку линий связи.

ГЛАВА 18 Дополнительные функции маршрутизаторов IP-сетей

Основными функциями IP-маршрутизатора являются создание таблицы маршрутизации и продвижение IP-пакетов на основе данных этой таблицы. Для выполнения этих функций маршрутизатор должен поддерживать протокол IP, рассмотренный в главе 16, а также протоколы маршрутизации, с которыми мы познакомились в главе 17. Помимо этих базовых функций современные IP-маршрутизаторы поддерживают ряд важных и более сложных функций, которые превращают IP-маршрутизаторы в гибкие и мощные многофункциональные устройства по обработке трафика. В этой главе мы рассмотрим наиболее важные из нетривиальных возможностей IP-маршрутизаторов, часто используемые администраторами сетей.

Маршрутизатор является пограничным устройством, соединяющим сеть с внешним миром. Поэтому естественно возложить на него *функции по защите сети* от внешних атак. Эти функции IP-маршрутизаторы выполняют путем фильтрации пользовательского трафика в соответствии с разнообразными признаками, передаваемыми в IP-пакетах: адресами отправителя и получателя, идентификатором типа протокола, вложенным в IP-пакет, идентификатором типа приложения, генерирующего этот трафик. Подобная функциональность предотвращает проникновение нежелательного трафика во внутреннюю сеть и снижает вероятность атаки на ее хосты. Важную роль в защите внутренних ресурсов сети играет *технология трансляции сетевых адресов* (NAT), которая позволяет скрыть от внешних пользователей реальные адреса, используемые хостами сети.

Сравнительно новым свойством IP-сетей является *поддержка параметров качества обслуживания* (QoS). Отдельные механизмы, необходимые для контроля и предотвращения перегрузок, IP-маршрутизаторы поддерживают на протяжении уже долгого времени, однако стандарты систем обеспечения QoS были разработаны для IP-сетей только в конце 90-х. Существуют две технологии поддержания параметров QoS для IP-сетей — это интегрированное и дифференцированное обслуживание. Первая обеспечивает качество обслуживания для отдельных потоков, а вторая разработана для агрегированных потоков, представляющих небольшое число классов трафика.

Еще одним очень перспективным направлением в развитии стека TCP/IP является *групповое вещание* (multicast). Помимо больших коммерческих перспектив эта технология увлекает исследователей своей сложностью. Действительно, немного задач можно сравнить по грандиозности с проблемой создания эффективного механизма одновременной доставки информации миллионам и миллиардам людей и устройств во всемирном масштабе.

Завершает главу рассмотрение особенностей *новой версии протокола IP — IPv6*. Мы наиболее подробно остановимся на модернизации схемы адресации, сделавшей ее более масштабируемой, а также на изменении формата заголовка IP, что позволило повысить пропускную способность сети за счет сокращения объема работы, выполняемой маршрутизаторами.

Фильтрация

Протоколы IP-маршрутизации создают таблицы маршрутизации, на основе которых любой узел составной сети может обмениваться информацией с любым другим узлом. Благодаря этому принципу дейтаграммных сетей каждый пользователь Интернета может получать доступ к любому публичному сайту.

Напомним, что в сетях, основанных не технике виртуальных каналов, взаимодействие произвольных узлов невозможно без предварительной процедуры установления между ними виртуального канала.

Однако такая всеобщая достижимость узлов и сетей не всегда отражает потребности их владельцев. Поэтому многие маршрутизаторы поддерживают развитые средства фильтрации пользовательского трафика, а также фильтрации объявлений протоколов маршрутизации, что позволяет дифференцированно управлять достижимостью узлов.

Фильтрация пользовательского трафика

Под **фильтрацией** понимается нестандартная обработка IP-пакетов маршрутизаторами, приводящая к отбрасыванию некоторых пакетов или изменению их маршрута.

Фильтрация пользовательского трафика маршрутизаторами аналогична по принципу действия фильтрации, выполняемой коммутаторами локальных сетей (см. главу 14).

Условия фильтрации маршрутизаторов обычно существенно сложнее и в них учитывается гораздо больше признаков, чем у коммутаторов локальных сетей. Например, это могут быть:

- IP-адреса источника и приемника;
- MAC-адреса источника и приемника;
- идентификатор интерфейса, с которого поступил пакет;
- тип протокола, сообщение которого несет IP-пакет (то есть TCP, UDP, ICMP или OSPF);
- номер порта TCP/UDP (то есть тип протокола прикладного уровня).

При наличии фильтра маршрутизатор сначала проверяет совпадение условия, описанного этими фильтром, с признаками пакета, и при положительной проверке выполняет над пакетом ряд нестандартных действий. Например, пакет может быть отброшен (drop); направлен к следующему маршрутизатору, отличающемуся от того, который указан в таблице маршрутизации; помечен, как вероятный кандидат на отбрасывание при возникновении перегрузки. Одним из таких действий может быть и обычная передача пакета в соответствии с записями таблицы маршрутизации.

Рассмотрим примеры фильтров, написанных на командном языке маршрутизаторов Cisco. Эти фильтры, называемые **списками доступа**, сегодня в IP-маршрутизаторах являются очень распространенным средством ограничения пользовательского трафика.

Наиболее простым является **стандартный список доступа**; в нем в качестве условия фильтрации учитывается только IP-адрес источника.

Общая форма такого условия выглядит следующим образом:

```
access-list номер_списка_доступа { deny | permit }
  { адрес_источника [ метасимволы_источника ] | any}
```

Стандартный список доступа определяет два действия с пакетом, который удовлетворяет описанному в фильтре условию: отбросить (deny) или передать для стандартной обработки в соответствии с таблицей маршрутизации (permit). Условием выбора того или иного действия в стандартном списке доступа является совпадение IP-адреса источника пакета с адресом источника, заданным в списке. Совпадение проверяется в том же стиле, что и при проверке таблицы маршрутизации, при этом **метасимволы** являются аналогом маски, но в несколько модифицированном виде. Двоичный нуль в поле метасимволов источника означает, что требуется совпадение значения этого разряда в адресе пришедшего пакета и в адресе, заданном в списке доступа. Двоичная единица означает, что совпадения в этом разряде не требуется. Практически, если вы хотите задать условие для всех адресов некоторой подсети, то должны использовать инвертированное значение маски этой подсети. Параметр any означает любое значение адреса — это просто более понятная и краткая форма записи значения 255.255.255.255 в поле метасимволов источника.

Пример стандартного списка доступа:

```
access-list 1 deny 192.78.46.0 0.0.0.255
```

Здесь:

- 1 — номер списка доступа;
- deny — действие с пакетом, который удовлетворяет условию данного списка доступа;
- 192.78.46.0 — адрес источника;
- 0.0.0.255 — метасимволы источника.

Этот фильтр запрещает передачу пакетов, у которых в старших трех байтах адреса источника имеется значение 192.78.46.0.

Список доступа может включать более одного условия. В этом случае он состоит из нескольких строк с ключевым словом access-list и одним и тем же номером списка доступа. Так, если мы хотим разрешить прохождение через маршрутизатор пакетов хоста 192.78.46.12, запрещая передачу пакетов одному из хостов сети 192.78.46.0/24, то список доступа будет выглядеть следующим образом:

```
access-list 1 permit 192.78.46.12 0.0.0.0
access-list 1 deny 192.78.46.0 0.0.0.255
access-list 1 permit any
```

Условия списка доступа проверяются по очереди, если какое-либо из них дает совпадение, то выполняется действие permit или deny, определенное в этом условии. После этого остальные условия списка уже не проверяются. Считается по умолчанию, что в конце каждого списка имеется неявное условие вида:

```
[access-list 1 deny any]
```

Однако, если вам все же требуется пропускать все пакеты, не определенные явно в условиях, необходимо добавить в последней строке условие:

```
access-list 1 permit any
```

Список доступа можно применять к любому интерфейсу маршрутизатора и в любом направлении: если список применяется с ключевым словом in, то он действует на входящие в интерфейс пакеты, а если с ключевым словом out — на выходящие. Например, написан-

ный нами список доступа 1 можно применить к некоторому интерфейсу для обработки входящего трафика, используя следующую команду:

```
access-group 1 in
```

Существуют также и более мощные типы списков доступа для маршрутизаторов Cisco, например, **расширенные списки доступа**. Общий формат этих списков следующий:

```
access-list номер_списка_доступа { deny | permit }
  { protocol | ключевое_слово_протокола }
  { адрес_источника [ метасимволы_источника ]] [ порт_источника ] | any }
  [ адрес_приемника [ метасимволы_приемника ]] [ порт_приемника ]
```

Пользуясь расширенными списками доступа, можно запретить прохождение во внутреннюю сеть предприятия FTP-пакетов. Как известно, служба FTP использует для приема запросов от клиентов протокол TCP с хорошо известным портом 21. Для этого в список доступа нужно включить условие:

```
access-list 102 deny TCP any 21 any
```

Затем можно применить его к интерфейсу маршрутизатора, к которому подключена внутренняя сеть, с ключевым словом `out`.

Администраторы корпоративных сетей из соображений безопасности[1] часто запрещают возможность трассировки извне внутренних хостов утилитой ping. Это делается с помощью условия:

```
access-list 101 deny ICMP any 192.78.46.8 0.0.0.0 eq 8
```

Как видно из условия, его синтаксис для протокола ICMP несколько отличается от общего синтаксиса расширенных списков доступа. Параметр `eq 8` означает, что запрещается передача ICMP-сообщений типа 8, соответствующего эхо-запросам, с помощью которых функционирует утилита ping.

Еще более гибким является язык фильтров программного маршрутизатора, работающего во многих версиях Unix. Синтаксис этого языка близок к синтаксису языка C, что позволяет строить весьма сложные логические конструкции с помощью условных инструкций `if`, `then`, `else`.

Необходимо отметить, что фильтрация пользовательского трафика может существенно замедлять работу маршрутизатора, так как обработка каждого пакета требует проверки дополнительных условий.

Для того чтобы не создавать слишком большую нагрузку на маршрутизатор и «не отвлекать» его от выполнения основных обязанностей, в фильтрах маршрутизаторов не используется информация о предыстории сеансов. Сколько бы ни было сложным условие фильтрации маршрутизатора, в нем учитываются только параметры *текущего* пакета и не могут учитываться параметры предыдущих пакетов, уже обработанных маршрутизатором. Это ограничение является главным отличием маршрутизаторов от брандмауэров, специальных программных систем, которые, используя информацию о предыстории сеансов, выполняют более качественную фильтрацию.

---

[1] См. об этом в главе 24.

Фильтрация маршрутных объявлений

Для контроля достижимости узлов и сетей можно, наряду с фильтрацией пользовательского трафика, *ограничить распространение объявлений протоколов маршрутизации*. Такая мера предотвращает автоматическое появление в таблице маршрутизации записей о некоторых сетях. Этот способ требует гораздо меньших затрат вычислительной мощности маршрутизатора, так как маршрутные объявления поступают на маршрутизатор гораздо реже, чем пользовательские пакеты.

Пусть, например, маршрутизаторы Cisco должны ограничить распространение маршрутных объявлений о какой-нибудь сети. Для этого нужно включить описание данной сети в стандартный список доступа, а затем применить к интерфейсу специальную команду с ключевым словом `distribute-list` (вместо `access-group`, как в случае фильтрации пользовательского трафика).

Например, если администратор не хочет, чтобы информация о внутренних сетях 194.12.34.0/24 и 132.7.0.0/16 предприятия распространялась по внешним сетям, ему достаточно написать следующий стандартный список доступа:

```
access-list 2 deny 194.12.34.0 0.0.0.255
access-list 2 deny 132.7.0.0 0.0.255.255
access-list 2 permit any
```

После этого достаточно применить его к интерфейсу с помощью команды

```
distribute-list 2 out serial 1
```

Стандарты QoS в IP-сетях

Технологии стека TCP/IP были разработаны для эластичного трафика, который достаточно терпим к задержкам и вариациям задержек пакетов. Поэтому основное внимание разработчиков протоколов TCP/IP было сосредоточено на обеспечении надежной передачи трафика с помощью TCP. Тем не менее для борьбы с перегрузками на медленных линиях доступа в IP-маршрутизаторы со временем были встроены многие механизмы QoS, в том числе механизмы приоритетных и взвешенных очередей, профилирования трафика и обратной связи. Однако эти механизмы использовались каждым сетевым администратором по своему усмотрению, без какой-либо стройной системы. И только в середине 90-х годов начались работы по созданию стандартов QoS для IP-сетей, на основе которых можно было бы создать систему поддержки параметров QoS в масштабах составной сети и даже Интернета.

В результате были разработаны две системы стандартов QoS для IP-сетей:

- система **интегрированного обслуживания** (Integrated Services, IntServ) ориентирована на предоставление гарантий QoS для потоков *конечных пользователей* (именно поэтому технология IntServ применяется в основном на периферии сети);
- система **дифференцированного обслуживания** (Differentiated Services, DiffServ) делает то же самое *для классов трафика*, и следовательно, ее предпочтительнее использовать на магистрали.

Обе системы опираются на все базовые элементы основанной на резервировании схемы поддержания параметров QoS, к которым относятся:

- кондиционирование трафика;
- сигнализация, обеспечивающая координацию маршрутизаторов;
- резервирование пропускной способности интерфейсов маршрутизаторов для потоков и классов;
- приоритетные и взвешенные очереди.

Ни одна из этих технологий не решает проблемы инжиниринга трафика, так как пакеты по-прежнему направляются вдоль пути с наилучшей метрикой, выбираемому стандартным протоколом маршрутизации без учета реальной загрузки каналов передачи данных.

Модели качества обслуживания IntServ и DiffServ

Направление **IntServ** начало разрабатываться в IETF еще в начале 90-х годов и было первым направлением, в рамках которого проблема обеспечения параметров QoS в сетях TCP/IP начала решаться систематически. Базовая модель IntServ предполагает интегрированное взаимодействие маршрутизаторов сети по обеспечению требуемого качества обслуживания *вдоль всего пути микропотока* между конечными компьютерами.

Ресурсы маршрутизаторов (пропускная способность интерфейсов, размеры буферов) распределяются в соответствии с QoS-запросами приложений в пределах, разрешенных политикой QoS для данной сети. Эти запросы распространяются по сети сигнальным **протоколом резервирования ресурсов** (Resource reSerVation Protocol, RSVP), который позволяет выполнять резервирование ресурсов для потоков данных.

Однако система IntServ обеспечения параметров QoS нашла довольно много противников, преимущественно среди поставщиков услуг Интернета (ISP). Дело в том, что при интегрированном обслуживании магистральные ISP-маршрутизаторы должны оперировать информацией о состоянии десятков тысяч микропотоков, проходящих через ISP-сети. Такая нагрузка на маршрутизаторы требует коренного пересмотра их архитектуры и, естественно, ведет к резкому повышению стоимости IP-сетей и предоставляемых ими услуг.

Поэтому в конце 90-х была создана другая более экономически эффективная технология QoS в IP-сетях, получившая название дифференцированного обслуживания (**DiffServ**). Она изначально была ориентирована на применение в пределах ISP-сетей, а конечные узлы, генерирующие микропотоки, в расчет не брались. Для технологии DiffServ поддержка параметров QoS начинается на пограничном маршрутизаторе ISP-сети, на который поступает большое количество микропотоков из сетей пользователей. Каждый пограничный маршрутизатор классифицирует и маркирует входящий трафик, разделяя его на небольшое число классов, обычно 3–4 (максимум — 8). Затем каждый маршрутизатор сети обслуживает классы трафика дифференцированно в соответствии с произведенной маркировкой, выделяя каждому классу определенное количество ресурсов. Резервирование ресурсов маршрутизаторов производится статически, чаще всего вручную администратором сети. Роль сигнального протокола играют метки принадлежности пакетов к тому или иному классу.

Ответственность за согласованное обслуживание трафика всеми маршрутизаторами сети несет администратор, так как он принимает решение, какие пропускную способность и величину буфера выделить каждому классу на каждом интерфейсе каждого маршрутизатора.

Модель DiffServ существенно снижает нагрузку на маршрутизаторы ISP-сети, так как требует хранить информацию о состоянии только небольшого количества классов. Кроме того, эта модель удобна для поставщиков услуг тем, что позволяет поддерживать параметры QoS автономно, только в пределах своих сетей. Однако за эти преимущества приходится платить, и прежде всего, отказом от гарантии сквозной поддержки параметров QoS. Даже если каждый поставщик услуг обеспечит дифференцированное обслуживание в своей сети, общая картина получится фрагментированной, так как за каждый фрагмент отвечает отдельный администратор, и согласование параметров резервирования остается исключительно субъективной процедурой, не поддерживаемой никакими протоколами.

Ведутся также работы по *комбинированному применению технологий IntServ и DiffServ*. Каждая технология в этих моделях работает в своей области, IntServ — в сетях доступа, где количество микропотоков относительно невелико, а DiffServ — в магистральных сетях. Еще одним компонентом, дополняющим DiffServ, является технология MPLS (см. главу 20).

Обе технологии (IntServ и DiffServ) опираются на одни и те же базовые механизмы QoS. В частности, в IP-маршрутизаторах для профилирования и формирования трафика применяется алгоритм ведра маркеров.

Алгоритм ведра маркеров

Алгоритм ведра маркеров позволяет оценить и ограничить среднюю скорость и величину пульсации потока пакетов. Этот алгоритм основан на сравнении потока пакетов с некоторым эталонным потоком. Эталонный поток представлен маркерами, заполняющими условное «ведро» маркеров (рис. 18.1).

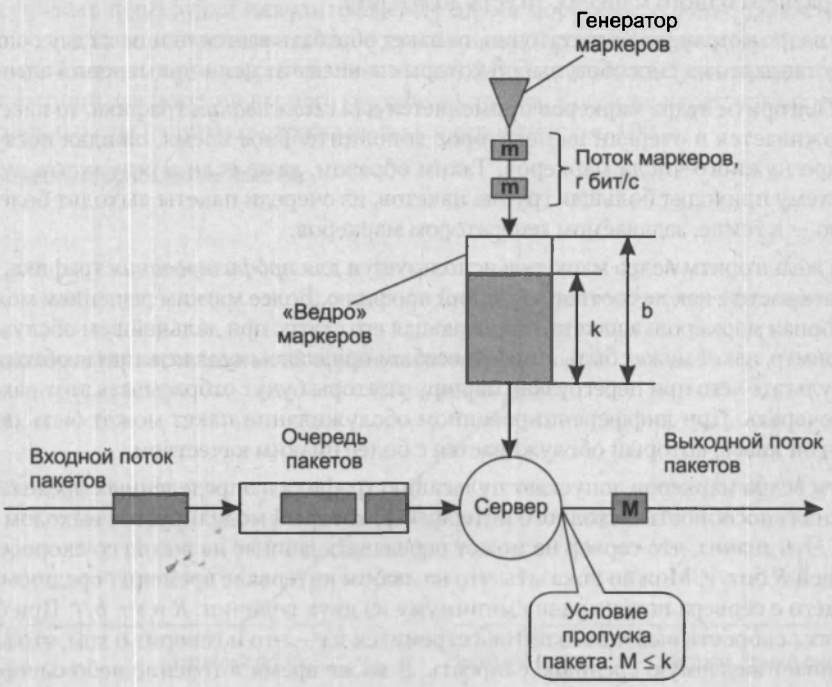


Рис. 18.1. Алгоритм ведра маркеров

Под маркером в данном случае понимается некий абстрактный объект, носитель «порции» информации, используемый для построения эталонного потока. Генератор маркеров периодически с постоянным интервалом w направляет очередной маркер в «ведро» с ограниченным объемом b байт. Все маркеры имеют одинаковый объем m байт, а генерация маркеров происходит так, что «ведро» заполняется со скоростью r бит/с. Нетрудно подсчитать, что $r = 8m/w$. Эта скорость r и является максимальной средней скоростью для трафика пакетов, а объем ведра соответствует максимальному размеру пульсации потока пакетов. Если ведро заполняется маркерами «до краев» (то есть суммарный объем маркеров в ведре становится равным b), то поступление маркеров временно прекращается. Фактически, ведро маркеров представляет собой счетчик, который наращивается на величину m каждые w секунд.

При применении алгоритма ведра маркеров профиль трафика определяется **средней скоростью** r и **объемом пульсации** b.

Сравнение эталонного и реального потоков выполняет сервер — абстрактное устройство, которое имеет два входа. Вход 1 связан с очередью пакетов, а вход 2 — с ведром маркеров. Сервер также имеет выход, на который он передает пакеты из входной очереди пакетов. Вход 1 сервера моделирует входной интерфейс маршрутизатора, а выход — выходной интерфейс.

Пакет из входной очереди продвигается сервером на выход только в том случае, если к моменту его поступления на сервер «ведро» заполнено маркерами до уровня не ниже M байт, где M — объем пакета.

При продвижении пакета из ведра удаляются маркеры общим объемом в M байт (с точностью до размера одного маркера, то есть до m байт).

Если же ведро заполнено недостаточно, то пакет обрабатывается одним из двух описанных далее нестандартных способов, выбор которых зависит от цели применения алгоритма.

- Если алгоритм ведра маркеров применяется для *сглаживания* трафика, то пакет просто задерживается в очереди на некоторое дополнительное время, ожидая поступления в ведро нужного числа маркеров. Таким образом, даже если в результате пульсации в систему приходит большая группа пакетов, из очереди пакеты выходят более равномерно — в темпе, задаваемом генератором маркеров.
- Если же алгоритм ведра маркеров используется для *профилирования* трафика, то пакет отбрасывается, как не соответствующий профилю. Более мягким решением может быть повторная маркировка пакета, понижающая его статус при дальнейшем обслуживании. Например, пакет может быть помечен особым признаком «удалять при необходимости», в результате чего при перегрузках маршрутизаторы будут отбрасывать этот пакет в первую очередь. При дифференцированном обслуживании пакет может быть переведен в другой класс, который обслуживается с более низким качеством.

Алгоритм ведра маркеров допускает пульсацию трафика в определенных пределах. Пусть пропускная способность выходного интерфейса, который моделируется выходом сервера, равна R. Это значит, что сервер не может передавать данные на выход со скоростью, превышающей R бит/с. Можно показать, что на любом интервале времени t средняя скорость исходящего с сервера потока равна минимуму из двух величин: R и $r + b/t$. При больших значениях t скорость выходного потока стремится к r — это и говорит о том, что алгоритм обеспечивает желаемую среднюю скорость. В то же время в течение небольшого времени t пакеты могут выходить из сервера со скоростью, большей r. Если $r + b/t < R$, то они

выходят из сервера со скоростью $r + b/t$, в противном случае интерфейс ограничивает эту скорость до величины R. Период времени t соответствует пульсации трафика. Эта ситуация наблюдается тогда, когда в течение некоторого времени пакеты не поступали в сервер, так что ведро полностью заполнилось маркерами (то есть времени, большего, чем b/r). Если после этого на вход сервера поступит большая группа пакетов, следующих один за другим, то эти пакеты будут передаваться на выход со скоростью выходного интерфейса R также один за другим, без интервалов. Максимальное время такой пульсации составляет $b/(R - r)$ секунд, после чего обязательно наступит пауза, необходимая для наполнения опустевшего ведра. Объем пульсации составляет $Rb/(R - r)$ байт. Из приведенного соотношения видно, что алгоритм ведра маркеров начинает плохо работать, если средняя скорость r выбирается близкой к пропускной способности выходного интерфейса. В этом случае пульсация может продолжаться очень долго, что обесценивает алгоритм.

Случайное раннее обнаружение

Механизм профилирования TCP-трафика, названный **случайным ранним обнаружением** (Random Early Detection, RED), разработан сообществом Интернета для предотвращения перегрузок на магистралях Интернета.

RED работает с протоколом TCP, используя свойство последнего, которое заключается в том, что при потерях пакетов источник трафика замедляет передачу пакетов в сеть. В алгоритме RED имеются два конфигурируемых порога уровня перегрузки (рис. 18.2). Когда уровень перегрузки не превышает первого (нижнего) порога, то пакеты не отбрасываются. Когда уровень перегрузки находится между двумя порогами, пакеты отбрасываются с линейно возрастающей вероятностью из диапазона от 0 до конфигурируемой величины (максимальной вероятности отбрасывания пакета). Максимальная вероятность отбрасывания действует при достижении второго (верхнего) порога. Когда же перегрузка превышает второй порог, пакеты начинают отбрасываться с вероятностью 100 %.

Рис. 18.2. Вероятность отбрасывания пакетов алгоритмом RED

В качестве показателя перегрузки используется вычисляемое среднее значение длины очереди пакетов, относящейся к определенному TCP-сеансу.

ПРИМЕЧАНИЕ

Заметим, что для UDP-трафика механизм RED неприменим, так как протокол UDP работает без установления логического соединения и, следовательно, потерь пакетов не замечает.

В том случае, когда нужно обеспечить разные параметры обратной связи для разных классов трафика, применяется **взвешенный алгоритм случайного раннего обнаружения** (Weighted Random Early Detection, WRED). Этот вариант алгоритма RED позволяет задавать для каждого класса трафика свои значения нижнего и верхнего порогов, а также вероятность отбрасывания пакетов. Обычно механизмы WRED и WFQ применяются совместно, обеспечивая надежную доставку TCP-трафика с гарантированной скоростью.

Интегрированное обслуживание и протокол RSVP

Интегрированное обслуживание основано на резервировании ресурсов маршрутизаторов вдоль пути следования потока данных от одного конечного узла (точнее, приложения) до другого (рис. 18.3). Приложение должно использовать соответствующий интерфейс API, чтобы передать запрос о резервировании ресурсов для определенного потока. Подобное резервирование является *однонаправленным*, так что если гарантированное качество обслуживания должно быть обеспечено для двустороннего обмена, потребуются две операции резервирования.

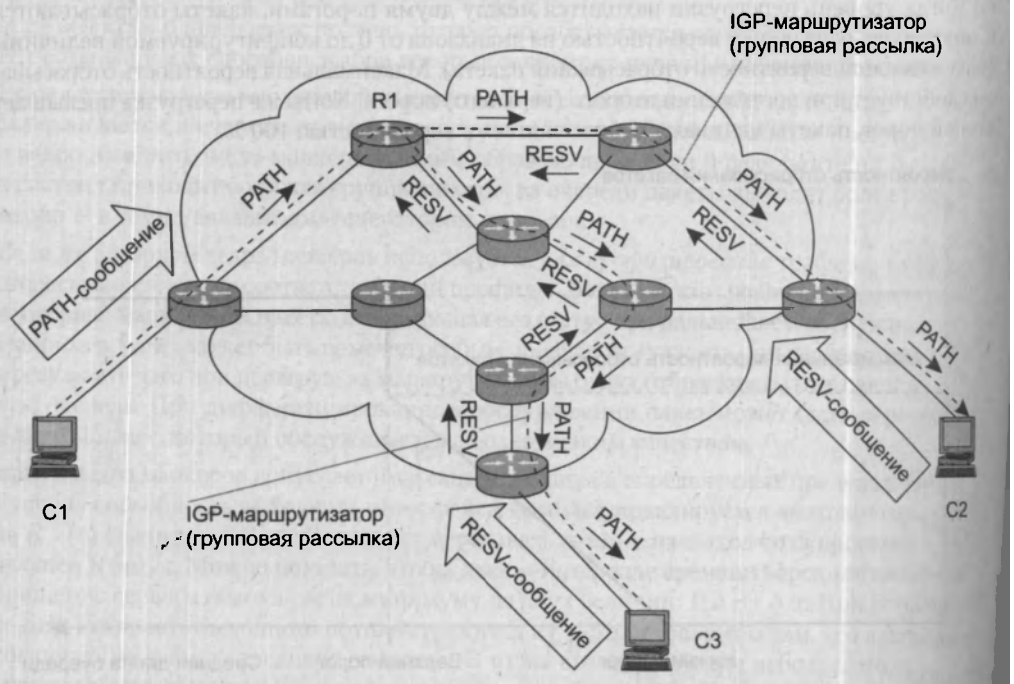


Рис. 18.3. Резервирование ресурсов по протоколу RSVP

Резервирование в модели IntServ выполняется с помощью уже упоминавшегося **протокола резервирования ресурсов** (RSVP). Это сигнальный протокол, во многом подобный *сигнальным протоколам телефонных сетей*. Однако специфика дейтаграммных пакетных сетей естественно накладывает свой отпечаток. Так, параметры коммутации в IP-сетях не являются атрибутом резервирования, потому что IP-пакеты в любом случае (при резервировании или без него) будут передаваться маршрутизаторами на основе записей таблицы маршрутизации.

Далее описана процедура резервирования необходимых ресурсов сети с помощью протокола RSVP, а в табл. 18.1 сведены воедино все упоминаемые в этом описании типы сообщений.

1. Источник данных (компьютер C1 на рис. 18.3) посылает получателям по уникальному или групповому (как на рисунке) адресу специальное **PATH-сообщение**, в котором указывает рекомендуемые параметры для качественного приема своего трафика: верхние и нижние границы пропускной способности, задержки и вариации задержки. Эти параметры составляют **спецификацию трафика источника**. PATH-сообщение передается маршрутизаторами сети в направлении ко всем указанным в групповом адресе получателям. В качестве параметров трафика применяются параметры алгоритма ведра маркеров, то есть средняя скорость и глубина ведра. Кроме того, дополнительно могут быть заданы максимально допустимая скорость и предельные размеры пакетов потока.
2. Каждый маршрутизатор, поддерживающий протокол RSVP, получив PATH-сообщение, фиксирует «состояние пути», которое включает предыдущий адрес источника PATH-сообщения, то есть последний по времени шаг в обратном направлении (ведущий к источнику). Это необходимо для того, чтобы ответ приемника прошел по тому же пути, что и PATH-сообщение.
3. После получения PATH-сообщения приемник отправляет в обратном направлении маршрутизатору, от которого он получил это сообщение, **запрос на резервирование ресурсов**, то есть **RESV-сообщение**. На рис. 18.3 показано два приемника, компьютеры C2 и C3. В дополнение к спецификациям трафика источника C1 (которые содержат параметры для качественного приема его трафика: верхние и нижние границы пропускной способности, задержки и вариации задержки) RESV-сообщение дополнительно включает **спецификацию запроса приемника**, в которой указываются требуемые *приемнику* параметры качества обслуживания, и **спецификацию фильтра**, которая определяет, к каким пакетам сеанса применять данное резервирование (например, по типу транспортного протокола и номеру порта). Вместе спецификации запроса и фильтра представляют собой **дескриптор потока**, для которого выполняется резервирование. Запрашиваемые параметры QoS в спецификации запроса могут отличаться от указанных в спецификации трафика. Например, если приемник решает принимать не все посылаемые источником пакеты, а только их часть (что указывается в спецификации фильтра), то ему нужна, соответственно, меньшая пропускная способность.
4. Каждый маршрутизатор, поддерживающий протокол RSVP вдоль восходящего пути, получив RESV-сообщение, проверяет, во-первых, имеются ли у маршрутизатора ресурсы, необходимые для поддержания запрашиваемого уровня QoS, а во-вторых, имеет ли пользователь право на резервирование ресурсов. Если запрос не может быть удовлетворен (из-за недостатка ресурсов или ошибки авторизации), маршрутизатор возвращает сообщение об ошибке отправителю. Если запрос принимается, то маршрутизатор посылает RESV-сообщение далее вдоль маршрута следующему маршрутизатору,

а данные о требуемом уровне QoS передаются тем механизмам маршрутизатора, которые ответственны за управление трафиком.

5. Прием маршрутизатором запроса на резервирование ресурсов означает также передачу параметров QoS на обработку в соответствующие блоки маршрутизатора. Конкретный способ обработки параметров QoS маршрутизатором в протоколе RSVP не описывается, но обычно она заключается в том, что маршрутизатор проверяет наличие свободной пропускной способности и емкости памяти для нового резервирования. При положительном результате проверки маршрутизатор запоминает новые параметры резервирования и вычитает их из счетчиков соответствующих свободных ресурсов.
6. Когда последний в обратном направлении маршрутизатор получает RESV-сообщение и принимает запрос, то он посылает подтверждающее сообщение узлу-источнику. При групповом резервировании учитывается тот факт, что в точках разветвления дерева доставки несколько резервируемых потоков сливаются в один. Так, в маршрутизаторе R1 в рассматриваемом примере сливаются RESV-сообщения от приемников C2 и C3. Если для всех резервируемых потоков запрашивается одинаковая пропускная способность, то она требуется и для общего потока, а если запрашиваются различные величины пропускной способности, то для общего потока выбирается максимальная.
7. После установления состояния резервирования в сети источник начинает отправлять данные, которые обслуживаются на всем пути к приемнику (приемникам) с заданным качеством обслуживания.

Таблица 18.1. Таблица сообщений протокола резервирования ресурсов (RSVP)

Типы сообщений	Содержание сообщений
PATH-сообщение от источника к приемнику	Спецификация трафика источника
Спецификация трафика источника	Рекомендуемые параметры для качественного приема своего трафика: верхние и нижние границы пропускной способности, задержки и вариации задержки, параметры алгоритма ведра маркеров, то есть среднюю скорость и глубину ведра, дополнительно могут быть заданы максимально допустимая скорость и предельные размеры пакетов потока
Спецификация фильтра	Определяет, к каким пакетам сеанса применять данное резервирование (например, по типу транспортного протокола и номеру порта)
Спецификация запроса приемника	Требуемые приемнику параметры качества обслуживания
Дескриптор потока	Спецификация фильтра плюс спецификация запроса приемника
RESV-сообщение – запрос на резервирование ресурсов	Спецификация трафика источника плюс дескриптор потока

Нужно подчеркнуть, что описанная схема обеспечивает резервирование только в одном направлении. Для того чтобы в рамках пользовательского сеанса данные передавались с заданным качеством обслуживания также и в обратном направлении, нужно, чтобы приемник и источник поменялись местами и выполнили RSVP-резервирование еще раз.

Для того чтобы параметры резервирования можно было применить затем к трафику данных, необходимо, чтобы RSVP-сообщения и пакеты данных следовали через сеть *одним и тем же маршрутом*. Это можно обеспечить, если передавать RSVP-сообщения на основе

тех же записей таблиц маршрутизации, которые применяются для пользовательского трафика.

ВНИМАНИЕ

Если для передачи RSVP-сообщений будет использоваться традиционная схема выбора маршрута в таблицах маршрутизации, то окажется упущенной возможность полноценного решения задач инжиниринга трафика, так как не все возможные маршруты будут задействованы для резервирования, а только кратчайший маршрут, выбранный в соответствии с некоторой метрикой протокола маршрутизации.

Резервирование можно отменить прямо или косвенно. Прямая отмена выполняется по инициативе источника или приемника с помощью соответствующих сообщений протокола RSVP. Неявная отмена происходит по тайм-ауту: состояние резервирования имеет срок жизни, как, например, и динамические записи в таблицах маршрутизации, и приемник по протоколу RSVP должен периодически подтверждать резервирование. Если же подтверждающие сообщения перестают поступать, то резервирование отменяется по истечении его срока жизни. Такое резервирование называется мягким.

Для протокола RSVP в настоящее время разработано большое количество расширений, которые делают его пригодным не только для работы в рамках архитектуры RSVP. Одними из наиболее важных являются расширения, относящиеся к инжинирингу трафика. Эти расширения применяются в технологии MPLS, рассматриваемой в главе 20.

Дифференцированное обслуживание

Дифференцированное обслуживание (DiffServ) опирается на ту же обобщенную модель QoS, что и интегрированное обслуживание, однако в качестве объектов обслуживания рассматриваются не отдельные потоки, а классы трафика.

Напомним, что **классом трафика** называется совокупность поступающих на обработку пакетов, обладающих общими признаками, например все пакеты голосовых приложений или все пакеты с MTU в определенных пределах.

В отличие от потока в классах трафика пакеты не различаются в зависимости от их маршрутов; это отличие иллюстрирует рис. 18.4. Так, маршрутизатор R1 относит все потоки, требующие приоритетного обслуживания и втекающие в его интерфейс i1, к одному классу, независимо от их дальнейшего маршрута. Маршрутизатор R2 оперирует уже другим составом приоритетного класса, так как в него вошли не все потоки интерфейса i1 маршрутизатора R1.

Обычно в сети DiffServ поддерживается дифференцированное обслуживание небольшого количество классов трафика, например двух (чувствительного к задержкам и эластичного) или трех (к первым двум прибавляется класс, требующий гарантированной доставки пактов с определенным минимумом скорости трафика). Небольшое количество классов определяет масштабируемость этой модели, так как маршрутизаторы не должны запоминать состояния каждого пользовательского потока. Высокая степень масштабирумости Diffserv обеспечивается также тем, что каждый маршрутизатор самостоятельно принимает решение о том, как он должен обслуживать тот или иной класс трафика, не согласуя свои

действия с другими маршрутизаторами. Такой подход назван *независимым поведением маршрутизаторов* (Per Hop Behavior, PHB). Так как в модели DiffServ маршруты пакетов не отслеживаются, то здесь не используется сигнальный протокол резервирования ресурсов, подобный протоколу RSVP в модели IntServ. Вместо этого маршрутизаторы сети выполняют статическое резервирование ресурсов для каждого из поддерживаемых сетью классов.

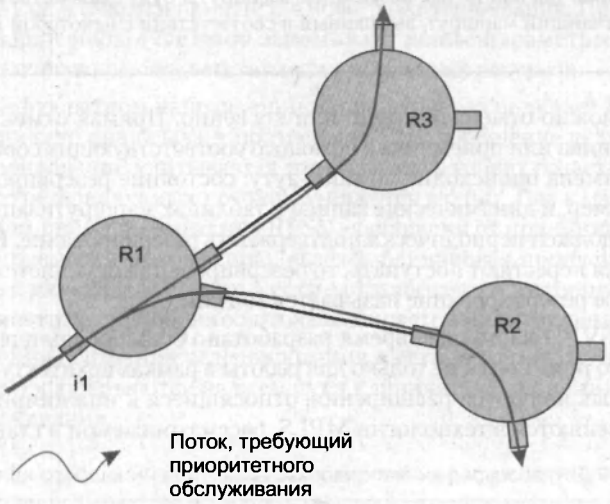


Рис. 18.4. В модели DiffServ объектами обслуживания являются классы трафика, а не потоки

В качестве признака принадлежности IP-пакета к определенному классу в DiffServ используется метка, переносимая в поле приоритета IP-пакета (ToS-байт), которое с появлением стандартов DiffServ было переопределено и названо DS-байтом. Как показано на рис. 18.5, DS-байт переопределяет значения битов ToS-байта, определенных ранее в соответствующих спецификациях (RFC 791, RFC 1122, RFC 1349).

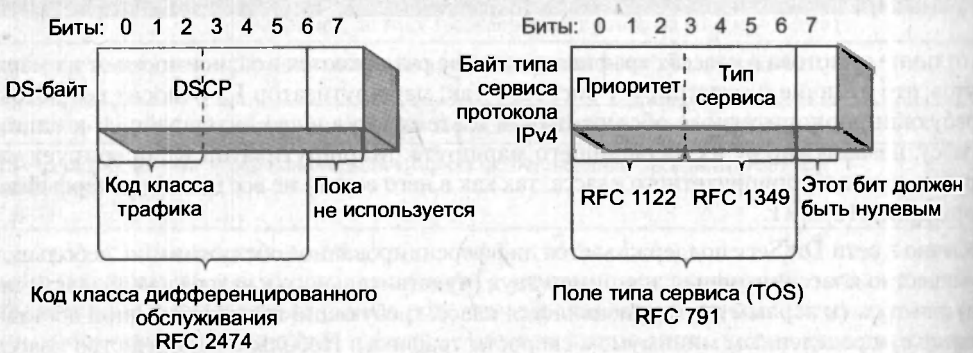


Рис. 18.5. Соответствие битов DS-байта битам поля типа сервиса

В настоящее время используются только старшие 6 битов DS-байта, причем только старшие три из них требуются для определения класса трафика (что дает не более 8-ми различ-

ных классов). Младший бит (из используемых шести) DS-байта обычно переносит признак IN – индикатор того, что пакет «вышел» из профиля трафика (аналогично признакам DE в технологии Frame Relay и CPL в технологии ATM). Промежуточные два бита обычно описывают различные варианты обслуживания пакетов внутри одного класса трафика.

Маршрутизатор, поддерживающий модель DiffServ, должен обеспечивать классификацию, маркирование, измерение и кондиционирование трафика, его обслуживание в приоритетной или взвешенной очереди и сглаживание.

Хотя маркировкой пакетов может заниматься каждый маршрутизатор сети, в модели дифференцированного обслуживания основным вариантом считается маркировка пакетов на границе сети, поддерживающей эту модель и находящейся под административным контролем одной организации. Такая сеть называется DiffServ-доменом. При выходе пакетов за пределы DiffServ-домена маркировка снимается, так что другой домен может назначить ее заново. Пограничные маршрутизаторы DiffServ-домена исполняют роль контрольно-пропускных пунктов домена, проверяя входящий трафик и определяя, имеет ли он право на дифференцированное обслуживание.

Модель DiffServ подразумевает существование соглашения об уровне обслуживания (SLA) между доменами с общей границей. Это соглашение определяет критерии политики предоставления сервиса, профиль трафика, а также гарантируемые параметры QoS. Ожидается, что трафик будет формироваться и сглаживаться в выходных точках домена в соответствии с SLA, а во входной точке домена будет кондиционироваться в соответствии с правилами политики. Любой трафик «вне профиля» (например, выходящий за верхние границы полосы пропускания, указанной в SLA) не получает гарантий обслуживания (или же оплачивается по повышенной стоимости в соответствии с SLA). Правила политики предоставления сервиса могут включать время дня, адреса источника и приемника, транспортный протокол, номера портов. В том случае, когда соблюдаются правила политики и трафик удовлетворяет оговоренному профилю, DiffServ-домен должен обеспечить при обслуживании этого трафика параметры QoS, зафиксированные в SLA.

На сегодняшний день в IETF разработано два стандарта пошагового продвижения пакетов для схемы PHB, которые представляют два разных варианта обслуживания.

- *Быстрое продвижение* (Expedited Forwarding, EF) характеризуется значением кода 10111 и представляет собой высший уровень качества обслуживания, обеспечивая минимум задержек и вариаций задержек. Любой трафик, интенсивность которого превышает указанную в профиле, отбрасывается.
- *Гарантированная доставка* (Assured Forwarding, AF) характеризуется четырьмя классами трафика и тремя уровнями отбрасывания пакетов в каждом классе — всего получается 12 различных типов трафика. Каждому классу трафика выделяются определенные минимум пропускной способности и размер буфера для хранения его очереди. Трафик, параметры которого превышают указанные в профиле, доставляется с меньшей степенью вероятности, чем трафик, удовлетворяющий условиям профиля. Это означает, что качество его обслуживания может быть понижено, но он не обязательно будет отброшен.

На основе этих **пошаговых спецификаций** и соответствующих соглашений об уровне обслуживания (SLA) могут быть построены **сервисы** для конечных пользователей «из конца в конец» — это EF-сервис и AF-сервис соответственно.

Основное назначение EF-сервиса — обеспечение качества обслуживания, сопоставимого с качеством обслуживания выделенных каналов, поэтому этот сервис называется также *сервисом виртуальных выделенных каналов*.

Поскольку EF-сервис допускает полное вытеснение другого трафика (например, при его реализации с помощью приоритетной очереди), то его реализация должна включать некоторые средства ограничения влияния EF-трафика на другие классы трафика, например, путем ограничения скорости EF-трафика на входе маршрутизатора по алгоритму ведра маркеров. Максимальная скорость EF-трафика и, возможно, величина пульсаций должны устанавливаться сетевым администратором.

Четыре класса AF-сервиса ориентированы на гарантированную доставку, но без минимизации уровня задержек пакетов, как это оговорено для EF-сервиса. Гарантированная доставка выполняется в том случае, когда входная скорость трафика не превышает отведенной данному классу минимальной пропускной способности. Реализация классов AF-трафика хорошо сочетается с EF-сервисом — EF-трафик может обслуживаться по приоритетной схеме, но с ограничением интенсивности входного потока. Оставшаяся пропускная способность распределяется между классами AF-трафика в соответствии с алгоритмом взвешенного обслуживания, который обеспечивает необходимую пропускную способность, но не минимизацию задержек. Реализация AF-сервиса предполагает (но не требует) взвешенного обслуживания для каждого класса с резервированной полосой пропускания, а также применения обратной связи (в форме RED).

Относительная простота определяет недостатки дифференцированного обслуживания. Главным недостатком является сложность предоставления количественных гарантий пользователям. Поясним это на примере сети, изображенной на рис. 18.6.

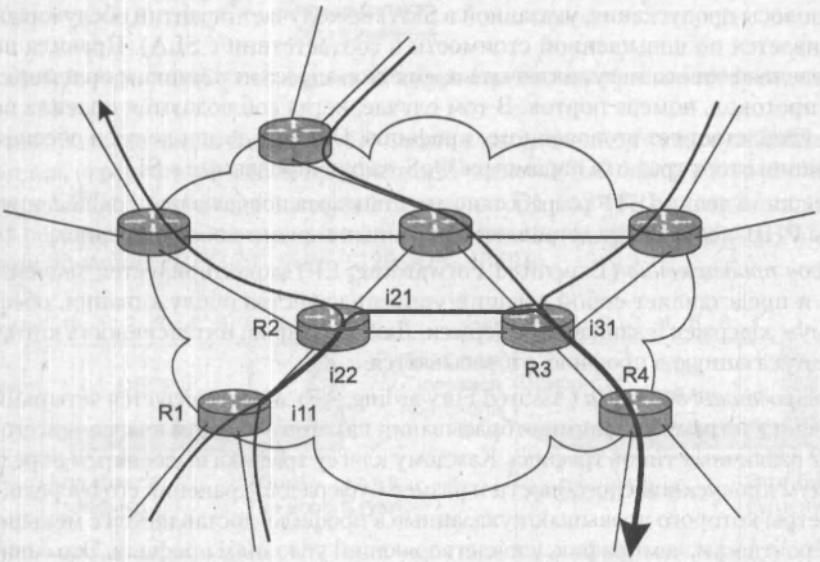


Рис. 18.6. Неопределенность уровня обслуживания в модели DiffServ

Обслуживание классов трафика подразумевает, что пограничные маршрутизаторы выполняют профилирование трафика без учета адреса назначения пакетов. Обычно для входных интерфейсов пограничных маршрутизаторов задается некоторый порог допустимой нагрузки для трафика каждого класса. Например, пусть наша сеть поддерживает трафик двух классов, реализуя особое обслуживание и обслуживание с максимальными усилиями,

причем порог для трафика с особым обслуживанием установлен в 20 % пропускной способности для каждого входного интерфейса каждого пограничного маршрутизатора. Кроме того, предположим для упрощения рассуждений, что все интерфейсы маршрутизаторов сети имеют одинаковую пропускную способность.

Несмотря на такое достаточно жесткое ограничение, интерфейсы маршрутизаторов сети оказываются под воздействием разной нагрузки. На рис. 18.6 для упрощения ситуации показаны только потоки, требующие особого обслуживания. Так, выходной интерфейс i11 маршрутизатора R1 обслуживает два таких потока и нагружен на 40 %, в то время как выходной интерфейс i21 маршрутизатора R2 — только один из них, так как второй поток уходит через другой выходной интерфейс. Выходной же интерфейс i31 маршрутизатора R3 перегружен, обслуживая три таких потока, так что его коэффициент использования равен 60 %. Учитывая факторы, влияющие на образование очередей (см. главу 7), мы знаем, что коэффициент использования является наиболее существенным фактором и значения в районе 50 % являются критическими. Поэтому в интерфейсе i31 возникают длинные очереди пакетов класса особого обслуживания, которые снижают качество такого обслуживания, так как приводят к длительным задержкам и их вариациям, а также потерям пакетов. Кроме того, страдает трафик класса обслуживания с максимальными усилиями, проходящий через этот интерфейс, так как ему достается только 40 % пропускной способности интерфейса.

Мы несколько утрировали картину, так как обычно интерфейсы магистральных маршрутизаторов являются более скоростными, чем пограничных, так что их коэффициент использования оказывается ниже, чем сумма коэффициентов использования входных интерфейсов, как в нашем примере. Для того чтобы снизить вероятность перегрузки внутренних интерфейсов магистральных маршрутизаторов и выходных интерфейсов пограничных маршрутизаторов, можно также уменьшить допустимый порог нагрузки входных интерфейсов трафиком особого обслуживания, например, до 5 %.

Однако все эти меры не дают гарантии, что все интерфейсы всех маршрутизаторов сети будут работать в нужном диапазоне значений коэффициента использования, а следовательно, обеспечивать заданное качество обслуживания. Для того чтобы дать такие гарантии, необходимо «улучшить» модель DiffServ и применять методы инжиниринга трафика, то есть контролировать не классы, а потоки трафика, в данном случае агрегированные. Под *агрегированным* понимается поток, состоящий из пакетов одного класса, имеющих общую часть пути через сеть. Эта общая часть не обязательно включает полный путь от входного интерфейса одного из пограничных маршрутизаторов до выходного интерфейса другого пограничного маршрутизатора. Достаточно, чтобы пакеты проходили хотя бы два общих интерфейса, чтобы считать их агрегированным потоком, как, например, в случае потока, проходящего через интерфейсы i11 и i22 (см. рис. 18.6).

Затем, зная путь прохождения каждого агрегированного потока через сеть, можно проверить, имеются ли достаточные ресурсы вдоль пути следования каждого потока, например, не превышают ли коэффициенты использования интерфейсов заданного порога. Для этого нужно провести профилирование с учетом адресов назначения пакетов. Однако реализация такого подхода в IP-сетях сталкивается с несколькими трудностями. Во-первых, в технологии Diffserv не предусмотрен сигнальный протокол, такой как, например, RSVP в технологии IntServ. Это означает, что все проверки наличия ресурсов у маршрутизаторов для каждого агрегированного потока нужно выполнять в автономном режиме, вручную или с помощью какого-то специального программного обеспечения. Во-вторых, для проведения

таких расчетов нужно знать пути потоков через сеть. Такие пути определяются таблицами маршрутизации, которые строятся протоколом маршрутизации, например RIP или OSPF (либо их комбинацией, если в сети используются несколько протоколов маршрутизации класса IGP), или вручную. Поэтому для ручного или автоматизированного расчета нужно знать таблицы маршрутизации всех маршрутизаторов сети и следить за их изменениями, а это непросто, учитывая, что отказы линий связи или маршрутизаторов приводят к перестройке таблиц. Нужно также учитывать, что маршрутизаторы могут применять методы балансировки нагрузки, разделяя агрегированный поток на несколько подпотоков, что также усложняет расчеты.

«Улучшенная» версия DiffServ, обеспечивающая учет адресов назначения, повышает качество услуг оператора связи, но вместе с тем усложняет саму идею метода, в основе которого лежит идея независимого обслуживания классов трафика каждым маршрутизатором сети.

Трансляция сетевых адресов

Маршрутизация в составной сети осуществляется на основе тех адресов назначения, которые помещены в заголовки пакетов. Как правило, эти адреса остаются неизменными с момента их формирования отправителем до момента поступления на узел получателя. Однако из этого правила есть исключения. Например, в широко применяемой сегодня технологии **трансляции сетевых адресов** (Network Address Translation, NAT) предполагается продвижение пакета во внешней сети (в Интернете) на основании адресов, отличающихся от тех, которые используются для маршрутизации пакета во внутренней (корпоративной) сети.

Причины подмены адресов

Одной из наиболее популярных причин использования технологии NAT является дефицит IP-адресов. Если по каким-либо причинам предприятию, у которого имеется потребность подключиться к Интернету, не удается получить у поставщика услуг необходимого количества глобальных IP-адресов, то оно может прибегнуть к технологии NAT. В этом случае для адресации внутренних узлов служат специально зарезервированные для этих целей **частные адреса**. Мы уже рассказывали о них в главе 15.

Для того чтобы узлы с частными адресами могли связываться через Интернет между собой или с узлами, имеющими глобальные адреса, необходимо использовать технологию NAT.

Технология NAT также оказывается полезной, когда предприятие из соображений безопасности желает скрыть адреса узлов своей сети, чтобы не дать возможности злоумышленникам составить представление о структуре и масштабах корпоративной сети, а также о структуре и интенсивности исходящего и входящего трафиков.

Традиционная технология NAT

Технология трансляции сетевых адресов имеет несколько разновидностей, наиболее популярная из которых — **традиционная технология трансляции сетевых адресов** — позволяет узлам из частной сети прозрачным для пользователей образом получать доступ к узлам внешних сетей. Подчеркнем, что в данном варианте NAT решается проблема организации только тех сеансов связи, которые *исходят* из частной сети. Направление сеанса в данном

случае определяется положением инициатора: если обмен данными инициируется приложением, работающем на узле внутренней сети, то сеанс называется исходящим несмотря на то, что в его рамках в сеть могут поступать данные извне[1].

Идея технологии NAT состоит в следующем. Пусть сеть предприятия образует тупиковый домен, узлам которого присвоены частные адреса (рис. 18.7). На маршрутизаторе, связывающем сеть предприятия с внешней сетью, установлено программное обеспечение NAT. Это NAT-устройство динамически отображает набор частных адресов {IP*} на набор глобальных адресов {IP}, полученных предприятием от поставщика услуг и присвоенных внешнему интерфейсу маршрутизатора предприятия.

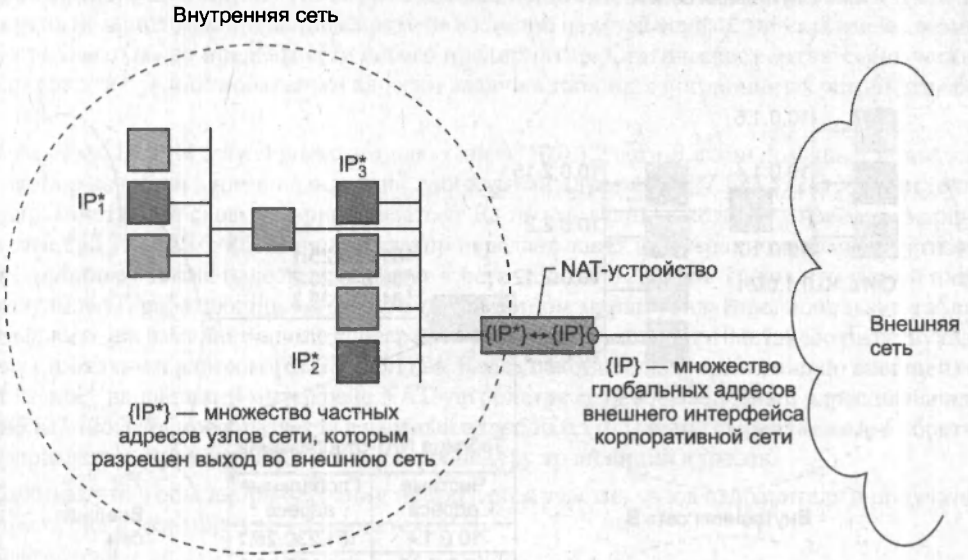


Рис. 18.7. Схема действия традиционной технологии NAT

Важным для работы NAT-устройства является правило распространения маршрутных объявлений через границы частных сетей. Объявления протоколов маршрутизации о внешних сетях «пропускаются» пограничными маршрутизаторами во внутренние сети и обрабатываются внутренними маршрутизаторами. Обратное утверждение неверно — маршрутизаторы внешних сетей не получают объявлений о внутренних сетях, объявления о них отфильтровываются при передаче на внешние интерфейсы. Поэтому внутренние маршрутизаторы «знают» маршруты ко всем внешним сетям, а внешним маршрутизаторам ничего не известно о существовании частных сетей.

Традиционная технология NAT подразделяется на технологии **базовой трансляции сетевых адресов** (Basic Network Address Translation, Basic NAT) и **трансляции сетевых адресов и портов** (Network Address Port Translation, NAPT). В технологии Basic NAT для отображения используются только IP-адреса, а в технологии NAPT — еще и так называемые транспортные идентификаторы, в качестве которых чаще всего выступают TCP- и UDP-порты.

[1] Традиционная технология NAT в виде исключения допускает сеансы обратного направления, заранее выполняя статическое взаимно однозначное отображение внутренних и внешних адресов для некоторого ограниченного набора узлов.

Базовая трансляция сетевых адресов

Если количество локальных узлов, которым необходимо обеспечить выход во внешнюю сеть, меньше или равно имеющегося количества глобальных адресов, то для каждого частного адреса гарантировано однозначное отображение на глобальный адрес. В каждый момент времени количество внутренних узлов, которые получают возможность взаимодействовать с внешней сетью, ограничивается количеством адресов в глобальном наборе. Понятно, что в такой ситуации целью трансляции является не столько решение проблемы дефицита адресов, сколько обеспечение безопасности.

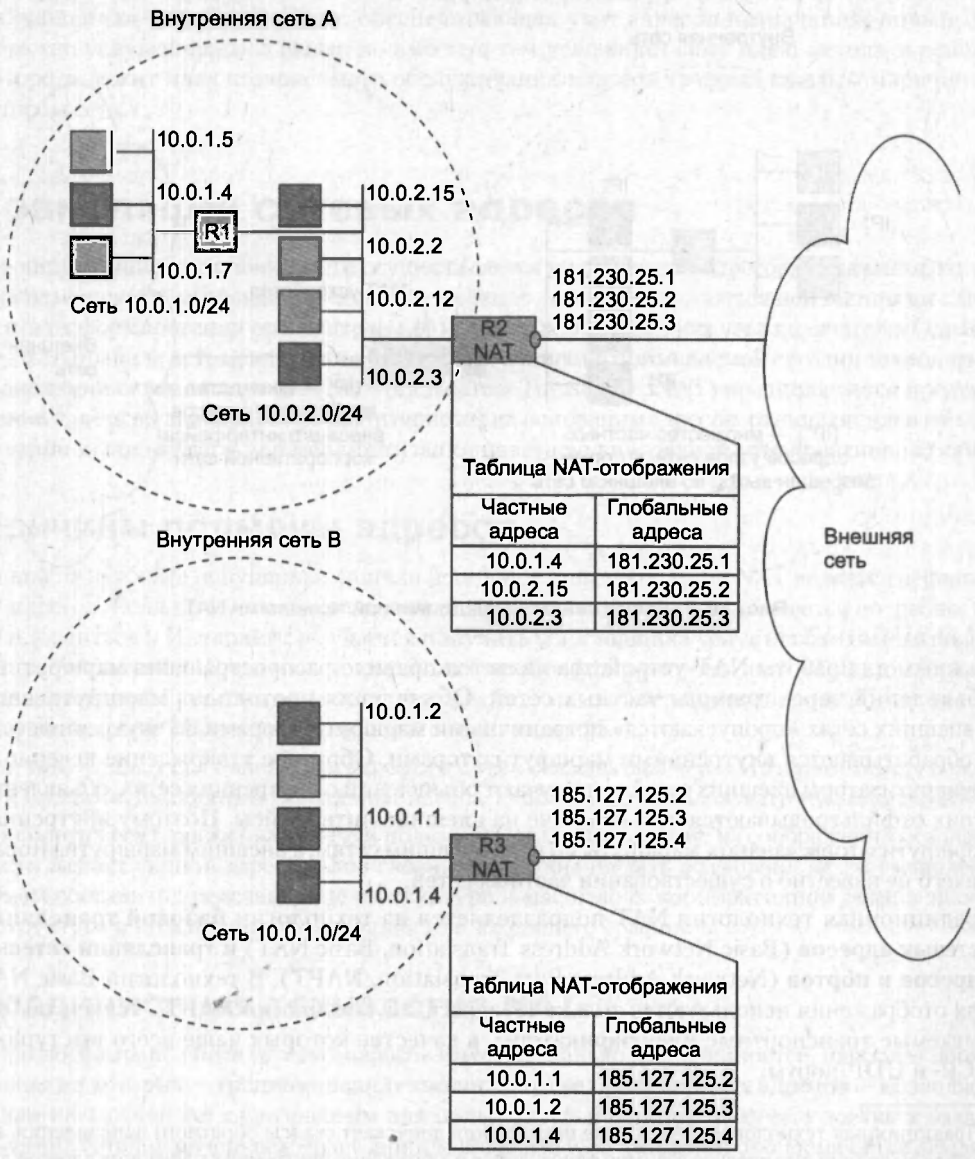


Рис. 18.8. Базовая трансляция сетевых адресов для исходящих сеансов

Частные адреса некоторых узлов могут отображаться на глобальные адреса *статически*. К таким узлам можно обращаться извне, используя закрепленные за ними глобальные адреса. Соответствие внутренних адресов внешним задается таблицей, поддерживаемой маршрутизатором или другим устройством (например, брандмауэром), на котором установлено программное обеспечение NAT.

В нескольких тупиковых доменах могут быть совпадающие частные адреса. Например, в сетях *A* и *B* на рис. 18.8 для внутренней адресации применяется один и тот же блок адресов 10.0.1.0/24. В то же время адреса внешних интерфейсов обеих сетей (181.230.25.1/24, 181.230.25.2/24 и 181.230.25.3/24 в сети *A* и 185.127.125.2/24, 185.127.125.3/24 и 185.127.125.4/24 в сети *B*) уникальны глобально, то есть никакие другие узлы в составной сети их не используют. В данном примере в каждой из сетей только три узла имеют возможность «выхода» за пределы сети своего предприятия. Статическое соответствие частных адресов этих узлов глобальным адресам задано в таблицах пограничных устройств обеих сетей.

Когда узел 10.0.1.4 сети *A* посылает пакет хосту 10.0.1.2 сети *B*, то он помещает в заголовок пакета в качестве адреса назначения глобальный адрес 185.127.125.3/24. Узел-источник направляет пакет своему маршрутизатору R1 по умолчанию, которому известен маршрут к сети 185.127.125.0/24. Маршрутизатор передает пакет на пограничный маршрутизатор R2, которому также известен маршрут к сети 185.127.125.0/24. Перед отправкой пакета модуль NAT, работающий на данном пограничном маршрутизаторе, используя таблицу отображения, заменяет в поле адреса источника частный адрес 10.0.1.4 соответствующим ему глобальным адресом 181.230.25.1/24. Когда пакет после путешествия по внешней сети поступает на внешний интерфейс NAT-устройства сети *B*, глобальный адрес назначения 185.127.125.3/24 преобразуется в частный адрес 10.0.1.2. Пакеты, передаваемые в обратном направлении, проходят аналогичную процедуру трансляции адресов.

Заметим, что в описанной операции не требуется участия узлов отправителя и получателя, то есть она прозрачна для пользователей.

Трансляция сетевых адресов и портов

Пусть некоторая организация имеет частную IP-сеть и глобальную связь с поставщиком услуг Интернета. Внешнему интерфейсу пограничного маршрутизатора R2 назначен глобальный адрес, а остальным узлам сети организации назначены частные адреса. NAPT позволяет *всем* узлам внутренней сети одновременно взаимодействовать с внешними сетями, используя единственный зарегистрированный IP-адрес. Возникает законный вопрос, каким образом внешние пакеты, поступающие *в ответ* на запросы из частной сети, находят узел-отправитель, ведь в поле адреса источника всех пакетов, отправляющихся во внешнюю сеть, помещается один и тот же адрес — адрес внешнего интерфейса пограничного маршрутизатора?

Для однозначной идентификации узла отправителя привлекается дополнительная информация. Если в IP-пакете находятся данные протокола UDP или TCP, то в качестве такой информации выступают номер UDP- или TCP-порта соответственно. Но и это не вносит полной ясности, поскольку из внутренней сети может исходить несколько запросов с совпадающими номерами портов отправителя, а значит, опять возникает вопрос об однозначности отображения единственного глобального адреса на набор внутренних адресов. Решение состоит в том, что при прохождении пакета из внутренней во внешнюю сеть каждой паре {внутренний частный адрес; номер TCP- или UDP-порта отправителя}

ставится в соответствие пара {глобальный IP-адрес внешнего интерфейса; назначенный номер TCP- или UDP-порта}. Назначенный номер порта выбирается произвольно, однако должно быть выполнено условие его уникальности в пределах всех узлов, получающих выход во внешнюю сеть. Соответствие фиксируется в таблице.

Эта модель при наличии единственного зарегистрированного IP-адреса, полученного от поставщика услуг, удовлетворяет требованиям по доступу к внешним сетям большинства сетей средних размеров.

На рис. 18.9 приведен пример, когда в тупиковой сети *A* используются внутренние адреса из блока 10.0.0.0. Внешнему интерфейсу маршрутизатора этой сети поставщиком услуг назначен адрес 181.230.25.1.

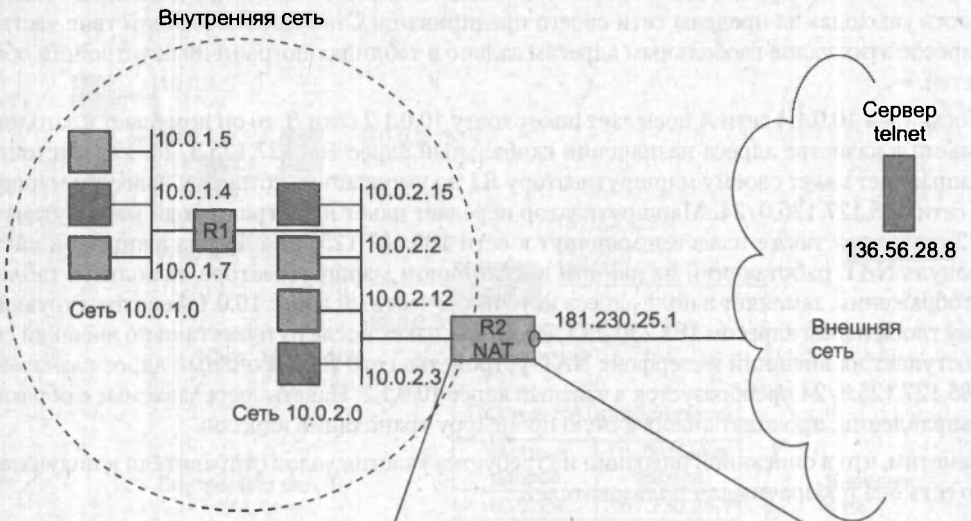


Таблица NAT-отображения

Частный адрес	Порт	Глобальный адрес	Назначенный порт
10.0.1.4	1245	181.230.25.1	3451
10.0.2.15	1245	181.230.25.1	3452
10.0.2.3	1045	181.230.25.1	3455

Рис. 18.9. Трансляция сетевых адресов и портов для исходящих TCP- и UDP-сеансов

Когда хост 10.0.1.4 внутренней сети посылает во внешнюю сеть пакет серверу telnet, то он в качестве адреса назначения использует его глобальный адрес 136.56.28.8. Пакет поступает маршрутизатору R1, который знает, что путь к сети 136.56.0.0/16 идет через пограничный маршрутизатор R2. Модуль NAPT маршрутизатора R2 транслирует адрес 10.0.1.4 и порт TCP 1245 источника в глобально уникальный адрес 181.230.25.1 и уникально назначенный TCP-порт, в приведенном примере – 3451. В таком виде пакет отправляется во внешнюю сеть и достигает сервера telnet. Когда получатель генерирует ответное сообщение, то он в качестве адреса назначения указывает единственный зарегистрированный глобальный адрес внутренней сети, являющийся адресом внешнего интерфейса NAPT-устройства. В поле номера порта получателя сервер помещает назначенный номер TCP-порта, взятый из поля порта отправителя пришедшего пакета. При поступлении ответного пакета на

NAPT-устройство внутренней сети именно по номеру порта в таблице трансляции выбирается нужная строка. По ней определяется внутренний IP-адрес соответствующего узла и действительный номер порта. Эта процедура трансляции полностью прозрачна для конечных узлов.

ВНИМАНИЕ

Заметьте, что в таблице имеется еще одна запись с номером порта 1245, такая ситуация вполне возможна: операционные системы на разных компьютерах независимо присваивают номера портов клиентским программам. Именно для разрешения такой неоднозначности и привлекаются уникальные назначенные номера портов.

В технологии NAPT разрешаются только исходящие из частной сети TCP- и UDP-сеансы. Однако возникают ситуации, когда нужно обеспечить доступ к некоторому узлу внутренней сети извне. В простейшем случае, когда служба зарегистрирована, то есть ей присвоен хорошо известный номер порта (например, WWW или DNS), и, кроме того, эта служба представлена во внутренней сети в единственном экземпляре, задача решается достаточно просто. Служба и узел, на котором она работает, однозначно определяются хорошо известным зарегистрированным номером порта службы.

Завершая рассмотрение технологии NAT, заметим, что помимо традиционной технологии NAT существуют и другие ее варианты, например технология двойной трансляции сетевых адресов, когда модифицируются оба адреса — и источника, и приемника (в отличие от традиционной технологии NAT, когда модифицируется только один адрес). Двойная трансляция сетевых адресов необходима, когда частные и внешние адресные пространства имеют коллизии. Наиболее часто это происходит, когда внутренний домен имеет некорректно назначенные публичные адреса, которые принадлежат другой организации. Подобная ситуация может возникнуть из-за того, что сеть организации была изначально изолированной и адреса назначались произвольно, причем из глобального пространства. Или же такая коллизия может быть следствием смены поставщика услуг, причем организация хотела бы сохранить старые адреса для узлов внутренней сети.

Групповое вещание

Групповое вещание, то есть доставка данных из одного источника сразу нескольким получателям, давно доказала свою полезность и необходимость в мире коммуникаций. Применяемая ранее только в радио и телевизионных сетях, в последние годы технология группового вещания все шире внедряется в компьютерные сети. В условиях, когда компьютерные сети постепенно становятся средством для передачи практически всех видов информации, без реализации направленного широковещания в них не обойтись.

Поэтому технологии группового вещания являются сегодня предметом интенсивного изучения в исследовательском сетевом сообществе. Ведущие производители сетевого оборудования и программных средств стремятся встроить поддержку группового вещания в свои продукты — маршрутизаторы, коммутаторы, операционные системы.

Наиболее актуальна проблема реализации группового вещания в Интернете. Из-за своей популярности и доступности Интернет представляет собой идеальную среду для массового распространения по подписке мультимедийной информации — аудиозаписей, видеофильмов, информационных дайджестов и т. п. Приложения, реализующие такого рода услуги,

требуют наличия механизма доставки одной и той же информации определенному кругу абонентов сети. В связи с этим технология группового вещания имеет очень хорошие перспективы, и уже сейчас некоторые провайдеры предлагают такие услуги своим клиентам.

Концепция группового вещания (multicast) нашла свое воплощение в ряде спецификаций протоколов группового взаимодействия в Интернете. В 1992 году появилась экспериментальная магистраль MBone, которая объединила 20 сетей через Интернет. С помощью этой магистрали была проведена первая аудиоконференция, которая позволила группе, образованной из членов IETF по всему миру, слышать то, что говорилось на собрании IETF в Сан-Диего.

Стандартная модель группового вещания IP

Основной целью группового вещания является создание эффективного механизма передачи данных от одного источника нескольким получателям. Для решения этой задачи могут использоваться несколько подходов, например индивидуальная рассылка, широковещательная рассылка, привлечение сервисов прикладного уровня.

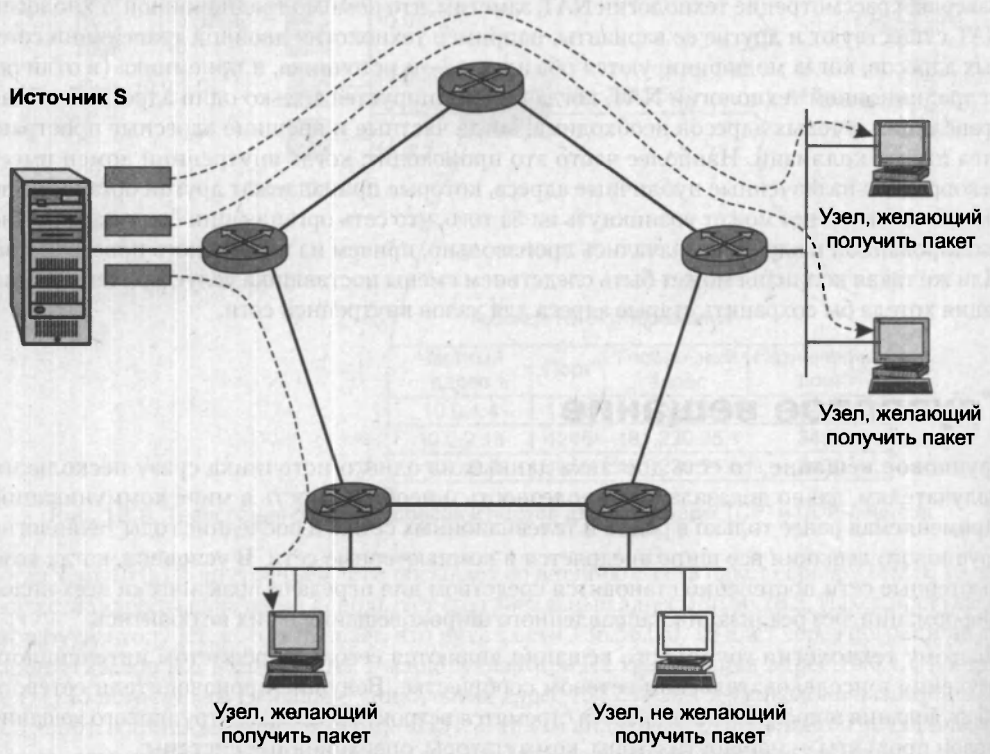


Рис. 18.10. Групповая доставка на основе индивидуальных адресов

При *индивидуальной рассылке* (unicast) на основе уникальных адресов источник данных, которые надо доставить некоторой группе узлов, генерирует их в количестве экземпляров,

равном количеству узлов-получателей, состоящих в данной группе (рис. 18.10). То есть передача по принципу «один ко многим» сводится к нескольким передачам «один к одному». Очевидно, что передача нескольких идентичных копий на участках, где маршруты к разным членам группы перекрываются (это особенно характерно для начальных участков), приводит к избыточному трафику.

При *широковещательной рассылке* (broadcast) станция направляет пакеты, используя широковещательные адреса (рис. 18.11). В этой схеме, для того чтобы доставить данные группе узлов-получателей, источник генерирует один экземпляр данных, но снабжает этот экземпляр широковещательным адресом, который диктует маршрутизаторам сети копировать эти данные и рассылать их всем конечным узлам независимо от того, «заинтересованы» узлы в получении этих данных или нет. В этом случае, как и в предыдущем, существенная доля трафика является избыточной.

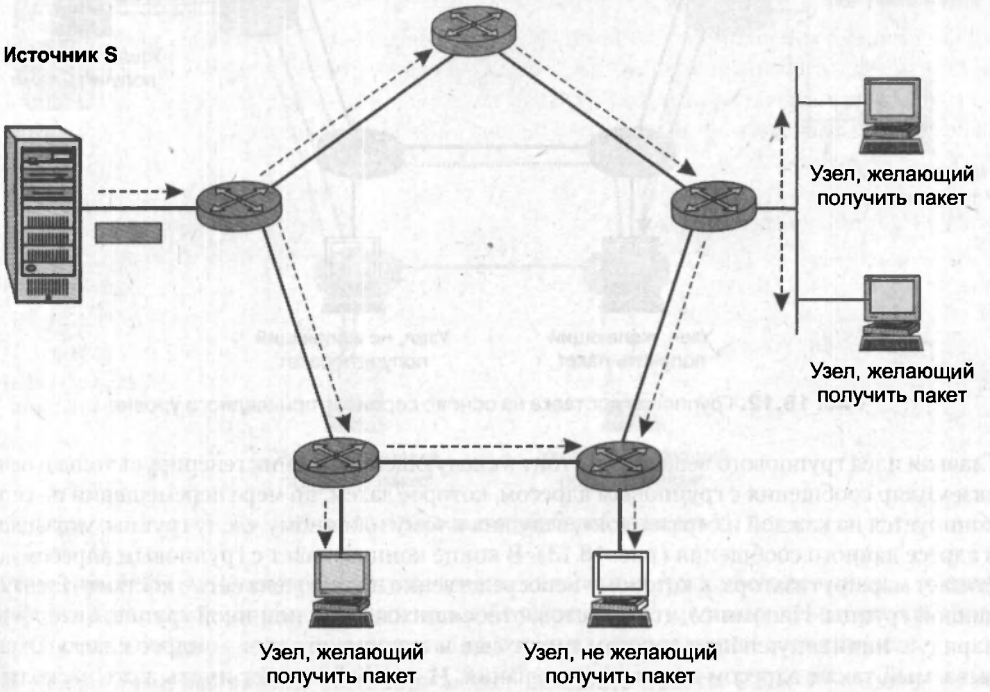


Рис. 18.11. Групповая доставка на основе широковещательного адреса

В случае *привлечения сервисов прикладного уровня* функции по обеспечению групповой доставки перекладываются на самих членов группы. То есть, как показано на рис. 18.12, источник генерирует один экземпляр данных и, используя индивидуальный адрес, передает данные одному из членов группы, который генерирует копию и направляет ее другому члену группы и т. д. Перемещение решения задачи с нижних транспортных уровней на прикладной уровень повышает суммарные накладные расходы сети на реализацию групповой доставки и делает этот механизм менее гибким.

Таким образом, традиционные механизмы доставки пакетов стека TCP/IP мало пригодны для поддержки группового вещания. В такой ситуации наиболее эффективным решением

является использование специально разработанного механизма группового вещания, ориентированного на сокращение избыточного трафика и накладных расходов сети.

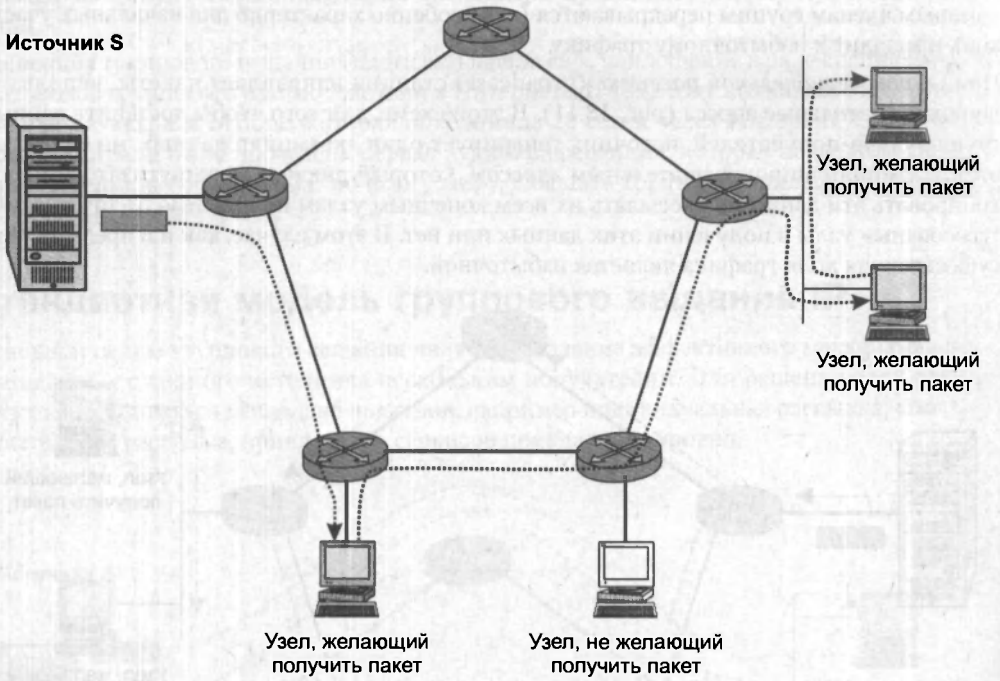


Рис. 18.12. Групповая доставка на основе сервисов прикладного уровня

Главная идея группового вещания состоит в следующем: источник генерирует только один экземпляр сообщения с групповым адресом, которое затем, по мере перемещения по сети, копируется на каждой из «развилок», ведущих к тому или иному члену группы, указанной в адресе данного сообщения (рис. 18.13). В конце концов, пакет с групповым адресом достигает маршрутизатора, к которому непосредственно подключена сеть с хостами-членами данной группы. Напомним, что у хостов, относящихся к той или иной группе, интерфейс наряду с индивидуальным адресом имеет еще и групповой адрес — адрес класса D, называемый также адресом группового вещания. Интерфейс может иметь даже несколько групповых адресов — по числу групп, в которых состоит данный хост.

Как и в случае обычной маршрутизации на базе индивидуальных адресов, маршрутизатор упаковывает пакет с групповым адресом в кадр канального уровня (той технологии, которая используется в данной локальной сети, например Ethernet), снабжая его групповым MAC-адресом, соответствующим групповому IP-адресу данного пакета[1]. Кадр с пакетом группового вещания поступает в локальную сеть, распознается и захватывается интерфейсами хостов, являющихся членами данной группы.

[1] Об отображении групповых IP-адресов на групповые MAC-адреса см. далее в разделе «Протокол IGMP».

При таком подходе данные рассылаются только тем узлам, которые заинтересованы в их получении. Функция репликации группового сообщения и продвижения копий в сторону членов группы возлагается на маршрутизаторы, для чего они должны быть оснащены соответствующими программно-аппаратными средствами. Такой режим экономит пропускную способность за счет передачи только того трафика, который необходим.

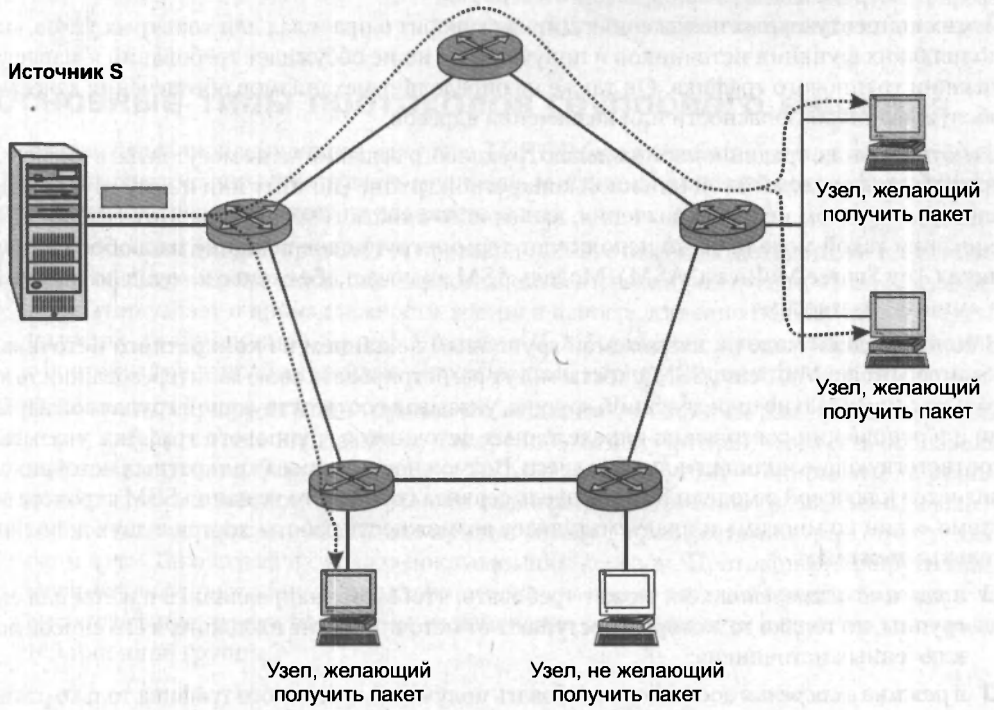


Рис. 18.13. Схема группового вещания

Стив Диринг (Steve Deering) — один из главных идеологов группового вещания — сформулировал несколько принципиальных положений, регламентирующих поведение конечных узлов сети, которые являются источниками и получателями группового трафика.

- *Дейтаграммный подход.* Источник может посылать пакеты UDP/IP в любое время без необходимости регистрировать или планировать передачи, реализуя сервис «по возможности».
- *Открытые группы.* Источники должны знать только групповой адрес. Они не должны знать членов группы и не обязательно должны быть членами той группы, которой они посылают данные. Группа может быть образована узлами, принадлежащими к разным IP-сетям и подсетям. Группа может иметь любое число источников данных.
- *Динамические группы.* Хосты могут присоединяться к группам или покидать группы без необходимости регистрации, синхронизации или переговоров с каким-либо централизованным элементом группового управления. Членство в группе является динамическим, поскольку хосты могут присоединиться к группе или выйти из группы в любой момент времени, к тому же они могут быть членами нескольких групп.

Из концепции открытых групп следует, что групповое вещание может быть организовано как по схеме «один ко многим», так и по схеме «многие ко многим».

Заметим также, что возможность быть источником никак не связана с членством хоста в той или иной группе. Источник группового вещания может начать передачу пакетов некоторой группе даже при условии, что во всем Интернете нет ни одного узла, который был бы заинтересован в этих данных.

В этих концептуальных положениях Диринг говорит о правилах для конечных узлов, выполняющих функции источников и получателей, но не обсуждает требований к маршрутизации группового трафика. Он также не определяет механизмов обеспечения качества обслуживания, безопасности или назначения адресов.

В соответствии с традиционной моделью группового вещания узлы могут делать заявки на трафик, направляемый той или иной конкретной группе (по тому или иному групповому адресу), при этом не имеет значения, каким источником генерируется этот трафик. Для описания такой модели часто используют термин **групповое вещание из любого источника** (Any Source Multicast, ASM). Модель ASM включает обе схемы: и «один ко многим», и «многие ко многим».

В более поздней модели, называемой **групповым вещанием из конкретного источника** (Source Specific Multicast, SSM), хосты могут регистрировать свою заинтересованность не только относительно определенной группы, указывая соответствующий групповой адрес, но и в отношении совершенно определенных источников группового трафика, указывая соответствующие индивидуальные адреса. Возможность запроса конкретных источников является ключевой в модели SSM. Модель сервиса группового вещания SSM строится по схеме «один ко многим» и предусматривает возможность работы хостов в двух дополнительных режимах:

- в *режиме исключения* хост может требовать, чтобы ему направлялись пакеты для его группы, но только те, которые поступают от источников, не входящих в его список исключенных источников;
- в *режиме включения* хост может требовать получение группового трафика только от тех источников, которые перечислены в списке включенных источников.

Адреса группового вещания

Ранее в главе 15, изучая типы IP-адресов, мы отмечали, что адреса IPv4 из диапазона 224.0.0.0–239.255.255.255 относятся к классу D и они зарезервированы для группового вещания.

Адреса из этого диапазона используются:

- для идентификации групп;
- для идентификации адресов источников группового вещания (в рамках модели SSM);
- для административных нужд при реализации группового вещания.

В общем случае адреса используются динамически, то есть если после остановки вещания источник снова начинает передачу, то он в общем случае может задействовать новый адрес группового вещания. Так называемые *хорошо известные* источники обычно наделяются постоянным групповым адресом.

Информацию о том, какие адреса уже закреплены для выполнения некоторой постоянной роли, а также о том, как использовать адресное пространство адресов класса D, дает документ RFC 3171 полномочной организации по цифровым адресам Интернета (Internet Assigned Numbers Authority, IANA).

Некоторые сведения из этого документа можно найти на сайте www.olifer.co.uk в разделе «Структурирование адресного пространства группового вещания».

Основные типы протоколов группового вещания

На основе описанной концепции для стека TCP/IP был разработан ряд протоколов, с помощью которых можно организовать групповое вещание с различной степенью эффективности. Эти протоколы делятся на две категории.

- В первую входит один протокол — протокол IGMP, с помощью которого, во-первых, хосты сообщают о своем «желании»[1] присоединиться к некоторой группе, во-вторых, маршрутизатор узнает о принадлежности хостов в непосредственно подключенных к нему подсетях к той или иной группе. Протокол IGMP работает в тесном взаимодействии с протоколами второй категории — протоколами маршрутизации группового вещания.
- **Протоколы маршрутизации группового вещания** необходимы для продвижения пакетов, несущих в себе информацию для групповых получателей, через сеть произвольной конфигурации. Эти протоколы — DVMRP, MOSPF, PIM — опираются на разные подходы, но в конечном итоге все они сводятся к построению графа, связывающего все хосты в определенной группе, причем между двумя хостами существует только один путь. Такой граф называют покрывающим деревом. Протоколы маршрутизации осуществляют постоянный мониторинг покрывающего дерева и время от времени отсекают те ветви дерева, которые из-за изменения состояния сети уже не ведут к членам той или иной группы.

Протокол IGMP

Протокол группового управления в Интернете (Internet Group Management Protocol, IGMP) был разработан в 1989 году для обеспечения более эффективной рассылки информации по IP-адресам, чем традиционные методы одноадресной и широковещательной передачи. Существует три версии IGMP: IGMPv1 (RFC 1112), IGMPv2 (RFC 2236) и IGMPv3 (RFC 3376).

Протокол IGMP используется исключительно при взаимодействии непосредственно связанных друг с другом маршрутизатора и хоста, когда последний выступает (или желает выступать) в роли получателя трафика группового вещания.

ПРИМЕЧАНИЕ

Источник не нуждается в протоколе IGMP. Любой компьютер, подключенный к Интернету, может стать источником группового вещания, при этом ему не требуется никакого дополнительного программного обеспечения, кроме того, которое включено в состав обычной реализации стека TCP/IP.

[1] Точнее, о «желании» приложения, выполняющегося на этом хосте, получать трафик, направляемый той или иной группе.

К основным функциям протокола IGMP относятся оповещение маршрутизатора о желании хоста быть включенным в группу и опрос членов группы.

Оповещение маршрутизатора о желании хоста быть включенным в группу. Чтобы стать получателем групповых данных, узел должен «выразить» свою заинтересованность маршрутизатору, к которому непосредственно подсоединена его сеть. Для этого хост должен установить взаимодействие с маршрутизатором по протоколу IGMP. Версия IGMP для хоста прямо зависит от типа операционной системы, установленной на хосте. Так, ранние версии Windows (Windows 95) поддерживали только версию IGMPv1, более поздние (Windows 2000) — версию IGMPv2, а начиная с Windows XP, поддерживается версия IGMPv3. Протоколы IGMPv2 и IGMPv3 поддерживаются во многих версиях Mac OS, Linux, Unix-подобных операционных системах.

Опрос членов группы. Для выполнения этой функции один из маршрутизаторов локальной сети выбирается доминирующим. Доминирующий маршрутизатор средствами протокола IGMP периодически опрашивает все системы (групповой адрес 224.0.0.1) в непосредственно присоединенных к нему подсетях, проверяя, активны ли члены всех известных ему групп. Остальные (не выбранные) маршрутизаторы прослушивают сеть, и если обнаруживают отсутствие сообщений-запросов в течение некоторого периода (обычно 250 секунд), то повторяют процедуру выбора нового доминирующего маршрутизатора.

В IGMPv2 определено три типа сообщений:

- *Запрос о членстве* (membership query). С помощью этого сообщения маршрутизатор пытается узнать, в каких группах состоят хосты в локальной сети, присоединенной к какому-либо его интерфейсу. Запрос о членстве существует в двух вариантах: в одном из них маршрутизатор делает общий запрос обо всех группах, в другом его интересует информация только о некоторой конкретной группе, адрес которой указывается в запросе.
- *Отчет о членстве* (membership report). Этим сообщением хосты отвечают маршрутизатору, который послал в сеть запрос о членстве. В сообщении содержится информация об адресе группы, в которой они состоят. Маршрутизатор, являясь членом всех групп, получает сообщения, направленные на любой групповой адрес. Для маршрутизатора, получающего ответные сообщения, важен только факт наличия членов той или иной группы (групп), а не принадлежность конкретных хостов конкретным группам. Этот факт будет использован другими маршрутизаторами сети для продвижения пакетов группового вещания в ту часть сети, за которую «отвечает» данный маршрутизатор. Отчет о членстве хост может послать не только в ответ на запрос маршрутизатора, но и *по собственной инициативе*, когда он пытается присоединиться к определенной группе. После такого сообщения хост может рассчитывать на то, что трафик для этой группы действительно будет доставляться в сеть, к которой этот хост принадлежит.
- *Покинуть группу* (leave group). Это сообщение хост *может* использовать, чтобы сигнализировать «своему» маршрутизатору о желании покинуть некоторую группу, в которой он до этого состоял. Получив это сообщение, маршрутизатор посылает специфический запрос о членстве членам только этой конкретной группы, и если не получает на него ответ (то есть это был последний хост в группе), то перестает передавать трафик группового вещания для этой группы. Слово «может» означает в данном случае, что хост может быть исключен из группы, просто не отвечая маршрутизатору на запрос о членстве (такой подход реализован в протоколе IGMPv1). Тогда маршрутизатор

будет продолжать передавать нежелательный трафик группового вещания до тех пор, пока не истечет некоторый период времени с момента поступления последнего отчета о членстве. Такой подход может значительно удлинить период скрытого нахождения хоста в состоянии выхода из группы, что снижает эффективность работы сети.

Сообщения с запросами о членстве посылаются маршрутизатором регулярно с некоторой частотой. На каждом из интерфейсов с установленными средствами IGMP маршрутизаторами поддерживаются кэш-таблицы групп. Кэш-таблица содержит список всех групп, в составе которых есть хотя бы один член. Для каждой строки таблицы установлен тайм-аут. Маршрутизатор регулярно посылает запросы (по умолчанию — каждые 125 секунд), чтобы проверить, что в каждой группе еще имеются члены. Если для некоторой группы ответ не поступает в течение установленного для нее тайм-аута, то соответствующая строка удаляется из кэш-таблицы, и маршрутизатор считает, что членов этой группы в сети больше нет.

Локальная сеть может иметь несколько хостов, заинтересованных в получении трафика одной и той же группы, но маршрутизатору достаточно подтверждения только от одного хоста для того, чтобы продолжить передавать трафик в сеть для этой группы. При использовании протокола IGMPv1 или IGMPv2 для ограничения числа ответов хостов на запрос маршрутизатора любой хост, состоящий в группе, вместо того чтобы немедленно ответить на запрос, сначала ждет в течение некоторого интервала времени, не появится ли в сети ответ какого-нибудь другого хоста. Если по истечении этого времени он так и не смог дождаться появления в сети ответа другого хоста, то он посылает маршрутизатору собственный отчет о членстве. (Если же используется протокол IGMPv3, то никаких пауз не устанавливается, и хосты сразу генерируют сообщения о членстве.)

Основываясь на информации, полученной с помощью IGMP, маршрутизаторы могут определять, в какие подключенные к ним сети необходимо передавать групповой трафик.

Все типы IGMP-сообщений имеют длину 8 байт и состоят из четырех полей. В зависимости от версии протокола IGMP назначение полей может несколько меняться. На рис. 18.14 показана структура сообщения для версии IGMPv2.

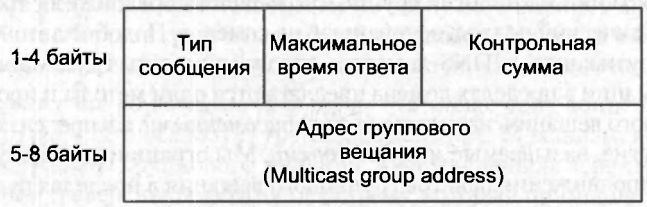


Рис. 18.14. Структура IGMP-сообщения

Поле максимального времени ответа используется хостами для вычисления времени задержки ответа. Время задержки выбирается случайным образом из интервала от нуля до значения, заданного в этом поле.

Заметим, что поле адреса группового вещания в IGMP-сообщении *не содержит* адрес назначения, оно несет в себе информацию, по-разному используемую в разных типах сообщений. Например, маршрутизатор, посылая запрос о членстве, помещает в этом поле нули, а хост в сообщениях «Отчет о членстве» и «Покинуть группу» помещает в это поле адрес группы, в которую он хочет вступить или которую он хочет покинуть соответственно.

ПРИМЕЧАНИЕ

Чтобы хост смог получать трафик группового вещания, недостаточно установить на нем протокол IGMP, с помощью которого хост может отправить сообщение своему маршрутизатору о желании присоединиться к группе. Помимо этого, надо сконфигурировать сетевой интерфейс хоста так, чтобы он стал захватывать из локальной сети кадры, несущие в себе пакеты группового вещания для той группы, к которой присоединился хост. Для этого необходимо настроить интерфейс на прослушивание определенного группового адреса канального уровня, соответствующего групповому IP-адресу. К сожалению, адресное пространство групповых IP-адресов в 32 раза объемнее пространства групповых MAC-адресов. То есть отображение этих двух адресных пространств оказывается далеко неоднозначным — на один и тот же групповой MAC-адрес отображается целый блок из 32 различных групповых IP-адресов. Следовательно, когда сетевой адаптер захватывает кадр, содержащий пакет группового вещания, существует значительная вероятность того, что этот пакет был направлен совсем другой группе. Однако эта ошибка скоро обнаруживается. Когда кадр передается вверх по стеку, протокол IP проверяет, совпадает ли групповой IP-адрес в поле адреса назначения инкапсулированного пакета с групповым IP-адресом данного интерфейса. (Отметим, что ни групповые IP-адреса, ни групповые MAC-адреса никогда не используются в качестве адресов отправителя.)

Принципы маршрутизации трафика группового вещания

Среди принципов маршрутизации трафика группового вещания можно отметить:

- маршрутизацию на основе доменов;
- учет плотности получателей группового трафика;
- два подхода к построению маршрутного дерева;
- концепцию продвижения по реверсивному пути.

Маршрутизация на основе доменов. Значительный объем хранимой и передаваемой по сети служебной информации, используемой для поддержания группового вещания, стал фактором, ограничивающим масштабируемость данной технологии. Для улучшения масштабируемости разработчики технологии группового вещания предложили традиционный для Интернета иерархический подход, основанный на доменах. Подобно автономным системам (доменам маршрутизации) и DNS-доменам, вводятся **домены группового вещания**. Для доставки информации в пределах домена предлагаются одни методы и протоколы маршрутизации группового вещания, называемые *внутридоменными*, а в пределах многодоменной структуры — другие, называемые *междоменными*. Мы ограничимся в этом учебнике описанием средств продвижения пакетов группового вещания в пределах отдельного домена.

Учет плотности получателей группового трафика. Внутридоменные протоколы маршрутизации разделяются на два принципиально отличных класса:

- Протоколы **плотного режима** (Dense Mode, DM) разработаны в предположении, что в сетевом домене существует большое число принимающих узлов. Отсюда следует главная идея этих протоколов: сначала «затопить» сеть пакетами группового вещания по всем направлениям, останавливая продвижение пакетов, лишь когда находящийся на пути распространения трафика маршрутизатор явно сообщит, что далее ниже по потоку членов данной группы нет.
- Протоколы **разряженного режима** (Sparse Mode, SM) рассчитаны на работу в сети, в которой количество маршрутизаторов с подключенными к ним членами групп невелико

по сравнению с общим числом маршрутизаторов. В такой ситуации выгоднее не усекать некоторые пути распространения широковещательной рассылки, а использовать явные сообщения о необходимости присоединения подсетей к дереву рассылки.

В сети, использующей протокол класса SM, необходимо существование центрального элемента, обычно называемого **точкой рандеву**, или **встречи** (Rendezvous Point, RP). Точка встречи должна существовать для каждой имеющейся в сети группы и быть единственной для группы. Все узлы, заинтересованные в получении информации, предназначенной той или иной группе, должны регистрироваться в соответствующей точке встречи. Функции точки (или нескольких точек) встречи выполняет специально назначенный для этого маршрутизатор. В сети может быть несколько маршрутизаторов, играющих роли точек встречи.

ПРИМЕЧАНИЕ

Сейчас согласно общепринятому мнению предпочтительнее применять протоколы разряженного режима даже в тех ситуациях, когда плотность приемников достаточно высока.

Два подхода к построению маршрутного дерева. Как и при решении задачи маршрутизации на основе индивидуальных адресов, в сети с групповым вещанием маршрутизаторы анализируют топологию сети, пытаясь найти кратчайшие пути доставки данных от источников к получателям. При этом все протоколы маршрутизации группового вещания используют один из следующих двух подходов.

Для всех источников данной группы строится *единственный* граф связей, называемый **разделяемым деревом**. Этот граф связывает всех членов данной группы (точнее, все маршрутизаторы, к которым подключены локальные сети, имеющие в своем составе членов данной группы). Разделяемое дерево может включать также и необходимые для обеспечения связности маршрутизаторы, не имеющие в своих присоединенных сетях членов данной группы. Разделяемое дерево служит для доставки трафика всем членам данной группы от *каждого* из источников, вещающих на данную группу.

Для каждой группы строятся *несколько* графов по числу источников, вещающих на каждую из этих групп. Каждый такой граф, называемый **деревом с вершиной в источнике**, служит для доставки трафика всем членам группы, но только от *одного* источника

Концепция продвижения по реверсивному пути — это еще одна концепция, которую необходимо понять всем, кто реализует групповое вещание. Механизм, используемый для маршрутизации трафика группового вещания, в определенном аспекте является прямо противоположным (реверсивным) тому механизму, который применяется для продвижения обычного трафика на основе индивидуальных адресов.

Традиционная маршрутизация на основе индивидуальных адресов основывается на адресе назначения. То есть маршрутизаторы перемещают пакет с индивидуальным адресом по сети вперед, в направлении приемника.

Напротив, все пакеты с групповым адресом маршрутизаторы тиражируют и передают копии во все стороны — на все интерфейсы, кроме того, с которого этот пакет поступил. При этом в сложных сетях возможно образование петель — замкнутых маршрутов. Для правильной работы сети зациклившиеся пакеты необходимо распознавать и отбрасывать.

Петля не может возникнуть, если ли пакет прибыл от источника по ожидаемому пути, проложенному в соответствии с обычным алгоритмом маршрутизации, основанном на анализе

таблиц маршрутизации. А именно, маршрутизатор проверяет, является ли входной интерфейс, получивший групповой пакет, интерфейсом, через который пролегает кратчайший путь к источнику. Он делает это с помощью обычной таблицы маршрутизации, которая, как известно, содержит указания о рациональных путях ко всем сетям составной интерсети.

Проверка факта выполнения данного условия называется **продвижением по реверсивному пути** (Reverse Path Forwarding, RPF). Такое название объясняется тем, что эта процедура связана не столько с путями, ведущими вперед от текущего места нахождения пакета к пункту назначения, сколько с обратным (реверсивным) путем, который уже пройден пакетом от того места, где он находится сейчас, до источника. Только пакеты, которые прошли RPF-проверку, являются кандидатами для дальнейшего продвижения вдоль путей, ведущих к потенциальным получателям трафика группового вещания.

Концепция продвижения по реверсивному пути является главной при маршрутизации группового трафика независимо от того, какой протокол при этом использован. Механизм RPF применятся и в других вариантах организации группового вещания. Например, когда маршрутизатор пытается продвигать пакеты к точке встречи в сети, работающей в разряженном режиме, он выбирает интерфейс, от которого проходит кратчайший путь к точке встречи.

На этом этапе мы не предъявляли специфических требований к таблицам маршрутизации, на основании которых выполняется RPF-проверка. Некоторые протоколы, такие как DVMRP, строят собственную таблицу маршрутизации, в то время как, например, протокол PIM работает с таблицами маршрутизации, построенными другими протоколами.

Протокол DVMRP

Дистанционно-векторный протокол маршрутизации группового вещания (Distance Vector Multicast Routing Protocol, DVMRP), описанный в спецификации RFC 1075, может быть характеризован с самых общих позиций следующим образом:

- как следует из его названия, он основан на *дистанционно-векторном алгоритме* и, следовательно, обладает всеми особенностями, свойственными данному алгоритму;
- относится к классу *протоколов плотного режима*, использующих проверку *продвижения по реверсивному пути*;
- продвигает пакеты на основе *деревьев с вершинами в источниках*;
- является *протокольно зависимым* в том смысле, что для принятия решений о продвижении пакетов он не может использовать обычные (для индивидуальной рассылки) таблицы маршрутизации.

Протокол DVMRT был одним из первых протоколов продвижения группового трафика в исследовательской сети MBone. Групповая маршрутизация в ранней версии MBone была, в сущности, управляемой формой широковещания, когда пришедший пакет с групповым адресом передавался через все интерфейсы, кроме входного. Для борьбы с зацикливанием пакетов с групповыми адресами маршрутизаторы запоминали факт продвижения данного пакета и при его поступлении в следующий раз просто отбрасывали. Для сокращения бесполезного трафика в сети применялся протокол IGMP. С помощью этого протокола маршрутизаторы выясняли, имеются ли в непосредственно подключенных к нему сетях конечные узлы, принадлежащие к определенной группе, или нет. В том случае, когда маршрутизатор определял, что к некоторому интерфейсу подключена сеть, в которой нет членов группы, являющихся получателями группового пакета, он не передавал копию этого пакета чрез данный выходной интерфейс.

Однако такой прием не исключает полностью избыточный трафик в сети, так как маршрутизатор не может судить о целесообразности передач дальше непосредственно подключенных к нему подсетей. Маршрутизатор передает пакет следующему маршрутизатору даже в том случае, если у того в подключенных сетях нет членов группы и ни один маршрут, проходящий через него, не ведет к сетям, в состав которых входят члены группы. На рис. 18.15 зачеркнуты избыточные маршруты группового трафика от узла *S*, по которым передаются пакеты туда, где нет ожидающих их получателей.

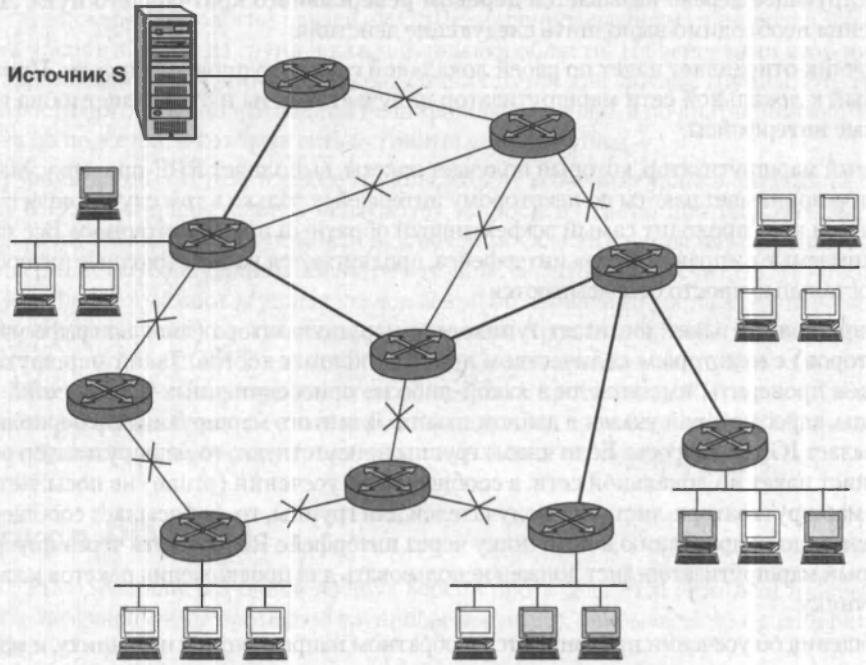


Рис. 18.15. Управляемое широковещание

Чтобы модернизировать протокол DVMPR, понадобилось несколько лет дополнительных усилий.

Цель модернизации состояла в распространении группового трафика от источника к получателям таким образом, чтобы пакеты продвигались только по тем путям, которые *единственным* и *кратчайшим* образом соединяли источник с каждым получателем.

Такие пути образуют *дерево с вершиной в источнике*, соединяющее кратчайшими путями все маршрутизаторы, к которым непосредственно подключены локальные сети, содержащие получателей данной группы, с маршрутизатором, к которому непосредственно подсоединена сеть, содержащая источник. Дерево для источника *S* и членов показанной на рисунке группы образуется оставшимися (незачеркнутыми) путями.

ПРИМЕЧАНИЕ

Для построения деревьев с вершиной в источнике пригодны различные алгоритмы, в частности один из таких алгоритмов, разработанный и стандартизованный IEEE для мостов локальных сетей под названием STA, мы рассмотрели ранее в главе 14.

Дальнейший прогресс в области алгоритмов маршрутизации для группового вещания был связан с разработкой алгоритма *плотного режима*, получившего название **широковещание и усечение** (broadcast-and-prune). Этот алгоритм рассчитан на то, что сети плотно «населены» членами различных групп, поэтому ситуация, когда в какой-либо подсети члены группы отсутствуют, считается редкой и отрабатывается особо. В этом «особом» случае маршрутизатор, обнаруживший подсеть, не содержащую членов группы, оповещает об этом другие маршрутизаторы и инициирует процедуру усечения избыточных маршрутов.

Результирующее дерево называется **деревом реверсивного кратчайшего пути**. Для его построения необходимо выполнить следующие действия:

1. Источник отправляет пакет по своей локальной сети с групповым адресом. Присоединенный к локальной сети маршрутизатор получает пакеты и отправляет их на все выходные интерфейсы.
2. Каждый маршрутизатор, который получает пакеты, выполняет RPF-проверку. Маршрутизатор принимает пакеты по некоторому интерфейсу только в том случае, если считает, что через него проходит самый эффективный обратный путь к источнику. Все пакеты, принимаемые с «правильного» интерфейса, продвигаются на все выходные интерфейсы. Все остальные просто отбрасываются.
3. В конце концов пакет достигает тупикового маршрутизатора (лист на графе маршрутизаторов) с некоторым количеством присоединенных хостов. Такой маршрутизатор должен проверить, имеются ли в какой-либо из присоединенных к нему сетей члены группы, адрес которой указан в данном пакете. Для этого маршрутизатор периодически рассылает IGMP-запросы. Если члены группы присутствуют, то маршрутизатор распространяет пакет по локальной сети, а сообщение об усечении (prune) не посылает. Если же у маршрутизатора-листа нет получателей для группы, то он посылает сообщение об усечении по направлению к источнику через интерфейс RPF, то есть через интерфейс, который маршрутизатор-лист должен использовать для продвижения пакетов к данному источнику.
4. Сообщения об усечении продвигаются в обратном направлении к источнику, и маршрутизаторы вдоль их пути фиксируют состояние усечения для интерфейса, через который получено сообщение об усечении.

Как уже было сказано, протоколы широковещания и усечения относятся к классу протоколов *плотного режима*, они эффективно работают, когда сеть плотно «населена» членами групп, так что далее по потоку имеются члены групп и поэтому целесообразно дальнейшее продвижение пакетов. Только когда приходит непосредственно сообщение об усечении, маршрутизатор перестает продвигать групповой трафик.

Главным недостатком протоколов плотного режима является то, что информация состояния для каждого источника должна храниться в каждом маршрутизаторе сети независимо от того, существуют ли члены групп вниз по потоку или нет. Если группа населена не очень плотно, то в сети нужно хранить значительный объем информации состояния и значительная часть пропускной способности может тратиться впустую.

Этот недостаток и стал толчком к разработке нового класса протоколов, названных протоколами *разряженного режима*. Вместо ориентации на существование большого количества членов группы, протоколы разряженного режима подразумевают наличие их в небольшом количестве, причем рассеянном по сети, как это часто и бывает в действительности. Мы рассмотрим два протокола «разряженного» режима — MOSPF и PIM-SM.

Протокол MOSPF

Протокол **MOSPF** (Multicast extensions to OSPF — расширения протокола OSPF для группового вещания), описанный в спецификации RFC 1584, опирается на обычные механизмы OSPF для поддержки группового вещания. MOSPF-маршрутизаторы добавляют к информации о состоянии связей, распространяемой по протоколу OSPF, данные о членстве в группах узлов в непосредственно присоединенных сетях. Эти данные рассылаются по сети в дополнительном сообщении о членстве в группе (group membership). В результате помимо топологии связей, MOSPF-маршрутизаторам становится известно о наличии членов каждой из групп в каждой подсети области. На основании этой информации маршрутизатор находит дерево кратчайших путей для каждой группы. Это позволяет распространять групповые пакеты не широковещательно, а по кратчайшим путям от источника до подсетей, в которых есть активные члены группы.

Для получения данных о том, в какие группы входят конечные узлы в связанных с ним подсетях, MOSPF-маршрутизатор использует запросы и ответы протокола IGMP. При каждом подключении узла к группе или исключении узла из группы маршрутизатор рассылает по сети новое сообщение о членстве в группе, так что можно считать, что протокол MOSPF задействует механизм явных уведомлений об изменении состава групп и поэтому относится к группе протоколов разряженного режима. Кроме того, известные положительные свойства протокола OSPF — устойчивое поведение при изменениях топологии сети, меньшие объемы служебного трафика по сравнению с протоколом RIP, а также возможность деления сети на области — полностью наследуются протоколом MOSPF, что делает его весьма привлекательным для применения в больших сетях.

Протокол PIM-SM

Протокол PIM-SM является одной из двух версий протокола **PIM** (Protocol Independent Multicast — независимое от протокола групповое вещание), описываемого в спецификации RFC 2362:

- версии плотного режима **PIM-DM** (Protocol Independent Multicast — Dense Mode);
- версии разряженного режима **PIM-SM** (Protocol Independent Multicast — Sparse Mode).

Эти версии существенно отличаются друг от друга способом построения и использования покрывающего дерева, но у них есть и одно общее свойство. Оно вынесено в название каждого из этих протоколов и означает независимость данного протокола от конкретных протоколов маршрутизации. Если DVMPR использует в своей работе механизмы RIP, а протокол MOSPF является расширением протокола OSPF, то протокол PIM может работать совместно с любым протоколом маршрутизации. Протокол PIM задействует готовые таблицы маршрутизации для продвижения групповых пакетов и служебных сообщений и для него не имеет значения, с помощью какого протокола маршрутизаторы строят эти таблицы.

Протокол PIM-DM похож на протокол DVMPR. Он, также являясь протоколом *плотного режима*, строит для доставки групповых пакетов *деревья с вершиной в источнике*, используя для этого проверки *продвижения по реверсивному пути* и технику *широковещания и усечения*. Основное отличие состоит в том, что PIM-DM применяет готовую таблицу маршрутизации, а не строит ее сам, как это делает DVMPR.

Главной особенностью протокола PIM-SM является то, что он рассчитан на работу в *разряженном режиме*, то есть он посылает групповые пакеты только по явному запросу получателя. Для доставки данных каждой конкретной группе получателей протокол PIM-SM строит одно *разделяемое дерево*, общее для всех источников этой группы (рис. 18.16).

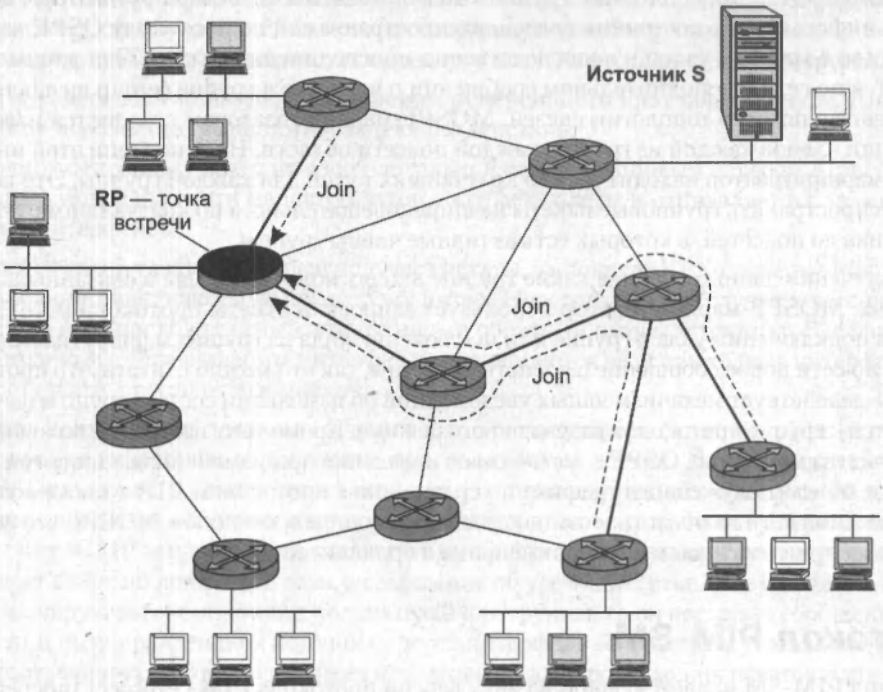


Рис. 18.16. Разделяемое дерево протокола PIM-SM

Вершина разделяемого дерева не может располагаться в источнике, так как источников может быть несколько. В качестве вершины разделяемого дерева используется специально выделенный для этой цели маршрутизатор, выполняющий функции *точки встречи* (RP). Все маршрутизаторы в пределах домена PIM-SM должны обладать согласованной информацией о расположении точки встречи. Различные группы могут иметь как одну и ту же, так и разные точки встречи.

Самым распространенным и возможно самым простым способом конфигурирования локальных (в пределах одного домена PIM-SM) точек встречи является назначение их *статически* среди множества маршрутизаторов данного домена. Это приводит к весьма определенной конфигурации и позволяет в дальнейшем легче находить ошибки, чем при других подходах.

Для получателей каждой конкретной группы и источников, вещающих на эту группу, маршрутизатор точки встречи является посредником, который связывает их между собой.

Процесс доставки протоколом PIM-SM группового трафика от источника к получателям, принадлежащим некоторой группе, может быть представлен трехэтапным:

1. Построение разделяемого дерева с вершиной в точке встречи, которое описывает пути доставки групповых пакетов между точкой встречи и членами данной группы. Это дерево называют также **деревом точки встречи** (Rendezvous Point Tree, RPT).

2. Построение **дерева кратчайшего пути** (Shortest Path Tree, SPT), которое будет доставлять пакеты между источником данной группы и точкой встречи.
3. Построение *набора* SPT-деревьев, которые ради повышения эффективности будут использованы для доставки пакетов непосредственно между источником и каждым из получателей группы.

ПРИМЕЧАНИЕ

Порядок следования этапов не фиксирован. Например, источники группового вещания могут начать передачу до того, как появятся слушатели, заинтересованные в этом трафике, или дерево кратчайшего пути между источником и его слушателями может уже быть построенным, когда будет сделан новый запрос на присоединение к группе.

Рассмотрим работу протокола PIM-SM на простом примере. На рис. 18.17 показана однодоменная сеть, в которой протокол PIM-SM устанавливает связь между одним получателем *A* и одним источником *S*. Будем считать, что работа сети соответствует модели ASM (групповое вещание из любого источника), на всех узлах сети развернут протокол IGMP и все маршрутизаторы поддерживают протокол PIM-SM. Будем считать также, что точка встречи сконфигурирована статически: и источники, и получатели знают индивидуальный адрес точки встречи, роль которой в этой сети играет маршрутизатор *D*. Для оповещения узлов сети об адресе точке встречи имеется стандартный протокол автоматического оповещения, называемый **протоколом загрузки**.

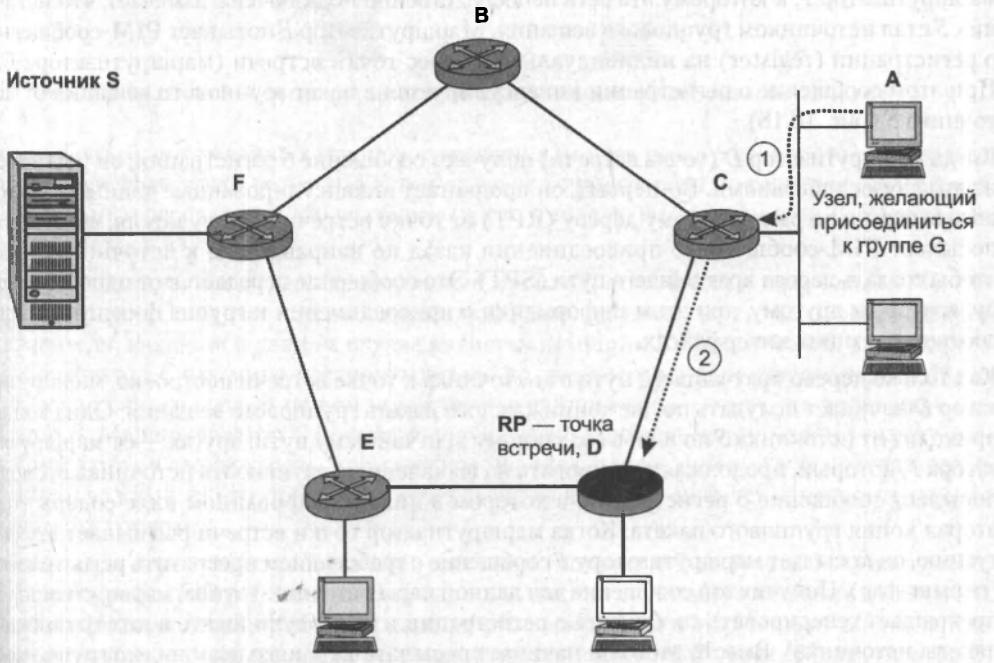


Рис. 18.17. Этап 1 — построение разделяемого дерева

Этап 1 — построение разделяемого RPT-дерева от получателя к точке встречи. Когда разделяемое дерево уже построено, трафик группового вещания передается от точки встречи

в направлении заинтересованных получателей. Однако процесс построения разделяемого дерева движется в обратном направлении — от получателей к точке встречи на основе пошагового (hop-by-hop) подхода.

Итак, пусть хост *A* решает присоединиться к группе *G*, по этой причине он посылает IGMP-сообщение отчета о членстве, содержащее адрес группы *G*, в локальную сеть, к которой он подключен. Это сообщение будет получено маршрутизатором *C*, через который данная локальная сеть подключена к другим сетям.

Маршрутизатор *C*, получив от хоста *A* это IGMP-сообщение, посылает *сообщение протокола PIM-SM* о присоединении (join) на индивидуальный адрес маршрутизатора *D*, выполняющего функции точки встречи. Это сообщение продвигается обычным образом на основе таблиц маршрутизации, построенных любыми протоколами маршрутизации. На всех промежуточных маршрутизаторах, расположенных вдоль пути от хоста-получателя к точке встречи, фиксируется состояние продвижения для данной группы. Каждый маршрутизатор добавляет интерфейс, принявший сообщение протокола PIM-SM о присоединении, к своему списку интерфейсов, через которые заинтересованным получателям может быть доставлен трафик группы, упомянутой в сообщении. В результате для данной группы формируется разделяемое дерево, и его корнем является точка встречи.

В нашем примере на данном этапе нет активных источников, поэтому данные группового вещания еще не поступают к точке встречи (см. рис 18.17).

Этап 2 — построение SPT-дерева от источника к точке встречи. Когда источник *S* становится активным и начинает посылать пакеты с групповым адресом в свою локальную сеть, маршрутизатор *F*, к которому эта сеть непосредственно подключена, замечает, что источник *S* стал источником группового вещания. Маршрутизатор *F* посылает PIM-сообщение о регистрации (register) на индивидуальный адрес точки встречи (маршрутизатора *D*). При этом сообщение о регистрации инкапсулируется в пакет группового вещания от источника *S* (рис. 18.18).

Когда маршрутизатор *D* (точка встречи) получает сообщение о регистрации, он реагирует на это двумя действиями. Во-первых, он продвигает инкапсулированные данные группового вещания по разделяемому дереву (RPT) от точки встречи до получателя, во-вторых, посылает PIM-сообщение о присоединении назад по направлению к источнику с тем, чтобы создать дерево кратчайшего пути (SPT). Это сообщение передается от одного маршрутизатора к другому, при этом информация о присоединении к группе фиксируется на соответствующих интерфейсах.

Как только дерево кратчайшего пути от источника к точке встречи построено, маршрутизатор *D* начинает получать по две копии каждого пакета группового вещания. Одна копия приходит от источника *S* по вновь созданному кратчайшему пути, другая — от маршрутизатора *F*, который, продолжая реагировать на выявленную активность источника *S*, снова посылает сообщение о регистрации, в котором в инкапсулированном виде содержится вторая копия группового пакета. Когда маршрутизатор точки встречи распознает эту ситуацию, он посылает маршрутизатору *F* сообщение с требованием прекратить регистрацию (register stop). Получив это сообщение для данной пары источник-группа, маршрутизатор *F* прекращает генерировать сообщения о регистрации и инкапсулировать в них групповые пакеты источника[1]. Вместо этого он начинает посылать их в исходном виде с групповым

[1] В дальнейшем маршрутизатор время от времени продолжит посылать одиночные сообщения о регистрации до тех пор, пока источник остается активным.

адресом, так как к этому моменту источник уже присоединился к дереву группы, и это присоединение зафиксировано на нужных маршрутизаторах.

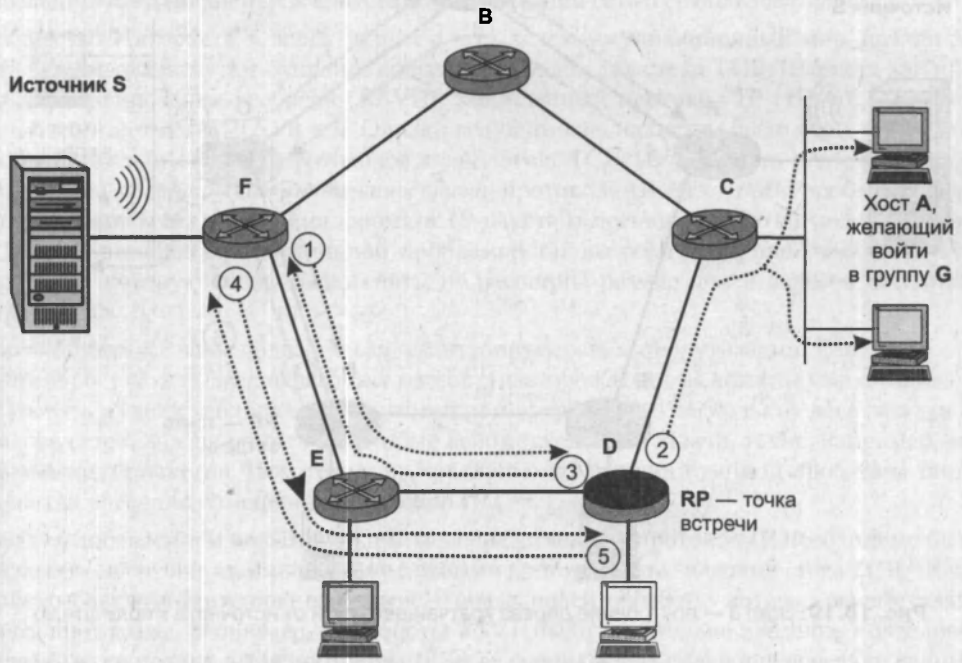


Рис. 18.18. Этап 2 — регистрация источника с построением дерева кратчайшего пути

Таким образом, поток данных группового вещания от источника *S* начинает передаваться по SPT-дереву до точки встречи, а затем далее от точки встречи по разделяемому дереву ко всем заинтересованным получателям (в том числе на маршрутизатор *C*, к которому подключен хост *A*).

Этап 3 — построение дерева кратчайшего пути от источника к получателю. Когда маршрутизатор *C* получает первый групповой пакет, он узнает из его заголовка IP-адрес отправителя, каковым в данном случае является источник *S*. На основании этого адреса маршрутизатор *C* пытается построить дерево кратчайшего пути непосредственно от источника до самого себя. В нашем примере кратчайший путь — это путь через маршрутизатор *B*. Маршрутизатор *C* посылает сообщение о присоединении маршрутизатору *B*, который затем, в свою очередь, посылает сообщение о присоединении маршрутизатору *F*. При этом каждый из них фиксирует интерфейс, на который он будет направлять пакеты для данной группы.

Теперь, когда дерево кратчайшего пути для пары (источник *S*, получатель *A*) построено, маршрутизаторы *F*, *B* и *C* начинают продвигать пакеты группового вещания вдоль него. Когда пакеты начинают прибывать на маршрутизатор *C*, он обнаруживает по две копии каждого пакета — одна приходит по новому кратчайшему пути через маршрутизатор *B*, другая по разделяемому дереву от маршрутизатора *D*. Чтобы прекратить дублирование, маршрутизатор *C* посылает PIM-сообщение об отсечении точки встречи (маршрутизатору *D*), который отсекает источник от разделяемого RPT-дерева (рис. 18.19).

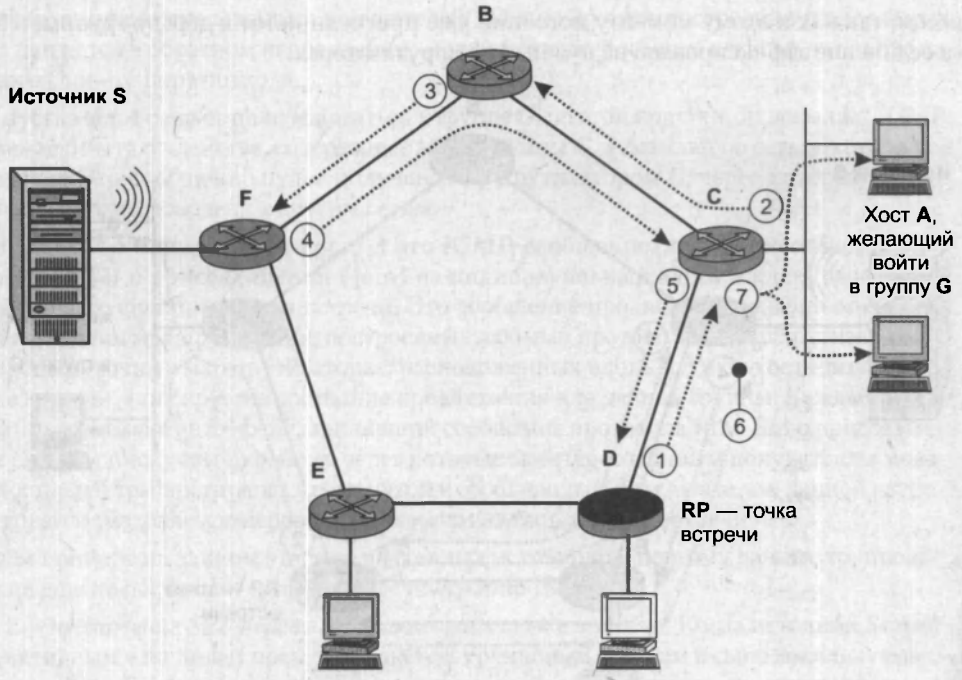


Рис. 18.19. Этап 3 — построение дерева кратчайшего пути от источника к получателю

С этого момента маршрутизатор *C* получает только по одной копии каждого пакета от источника *S* через свое отдельное дерево кратчайшего пути и передает его в локальную сеть, в которой находится получатель.

Для упрощения мы описали случай, когда в сети имеется только одна точка встречи и создается только одно разделяемое дерево. Однако технология допускает наличие в сети нескольких точек встречи. Решение о том, сколько в сети должно быть точек встречи и как их расположить, составляет предмет планирования сети и протоколом PIM не определяется.

Информацию об иерархическом подходе к организации группового вещания вы можете найти на сайте www.olifer.co.uk в разделе «Междоменное групповое вещание».

IPv6 как развитие стека TCP/IP

В начале 90-х годов стек протоколов TCP/IP столкнулся с серьезными проблемами. Именно в это время началось активное промышленное использование Интернета: переход к построению сетей предприятий на основе транспорта Интернета, применение веб-технологии для доступа к корпоративной информации, ведение электронной коммерции через Интернет, внедрение Интернета в индустрию развлечений (распространение видеофильмов, звукозаписей, интерактивные игры).

Все это привело к резкому росту числа узлов сети (в начале 90-х годов новый узел в Интернете появлялся каждые 30 секунд), изменению характера трафика и ужесточению требований, предъявляемых к качеству обслуживания сетью ее пользователей.

Сообщество Интернета, а вслед за ним и весь телекоммуникационный мир, начали решать новые задачи путем создания новых протоколов для стека TCP/IP, таких как протокол резервирования ресурсов (RSVP), защищенный протокол IP (IPSec), протокол коммутации меток (MPLS) и т. п. Однако ведущим специалистам было ясно, что только за счет добавления новых протоколов технологию TCP/IP развивать нельзя — нужно решиться на *модернизацию сердцевины стека*, протокола IP. Некоторые проблемы нельзя было решить без изменения формата IP-пакета и логики обработки полей заголовка IP-пакетов. Наиболее очевидной проблемой такого рода была проблема дефицита IP-адресов, которую невозможно снять, не расширив размер полей адресов источника и приемника.

Критике стала все чаще подвергаться масштабируемость маршрутизации. Дело в том, что быстрый рост сети вызвал перегрузку маршрутизаторов, которые должны уже сегодня обрабатывать в своих таблицах маршрутизации информацию о нескольких десятках тысяч номеров сетей, да еще решать некоторые вспомогательные задачи, такие, например, как фрагментация пакетов. Некоторые из предлагаемых решений данной проблемы также требовали внесения изменений в протокол IP.

Наряду с добавлением новых функций непосредственно в протокол IP, необходимо было обеспечить его тесное взаимодействие с новыми протоколами — членами стека TCP/IP, что также требовало добавления в заголовок IP новых полей, обработку которых осуществляли бы эти протоколы. Например, для работы RSVP было желательно введение в заголовок IP-поля метки потока, а для протокола IPSec — специальных полей для передачи данных, поддерживающих его функции обеспечения безопасности.

В результате сообщество Интернета после достаточно долгого обсуждения решило подвергнуть протокол IP серьезной переработке[1], выбрав в качестве основных целей модернизации:

- создание масштабируемой схемы адресации;
- сокращение объема работы, выполняемой маршрутизаторами;
- предоставление гарантий качества транспортных услуг;
- обеспечение защиты данных, передаваемых по сети.

Система адресации протокола IPv6

Новая (шестая) версия протокола IP (IPv6) внесла существенные изменения в систему адресации. Прежде всего, это коснулось увеличения разрядности адреса: вместо 4 байт IP-адреса в версии IPv4 в новой версии под адрес отведено *16 байт*. Это дает возможность пронумеровать огромное количество узлов:

340 282 366 920 938 463 463 374 607 431 762 211 456.

[1] В августе 1998 года были приняты пересмотренные версии группы стандартов, определяющих как общую архитектуру протокола IPv6 (RFC 2460), так и его отдельные аспекты, например, систему адресации (RFC 4291).

Масштаб этого числа иллюстрирует, например, такой факт: если разделить это теоретически возможное количество IP-адресов между всеми жителями Земли (а их сегодня примерно 6 миллиардов), то на каждого из них придется невообразимо, если не сказать бессмысленно большое количество IP-адресов — 5,7 × 1028! Очевидно, что такое значительное увеличение длины адреса было сделано не только и даже не столько для снятия проблемы дефицита адресов.

Главной целью изменения системы адресации было не механическое увеличение адресного пространства, а повышения эффективности работы стека TCP/IP в целом.

Вместо прежних двух уровней иерархии адреса (номер сети и номер узла) в IPv6 имеется 4 уровня, из которых три уровня используются для идентификации сетей, а один — для идентификации узлов сети. В новой версии не поддерживаются классы адресов (A, B, C, D, E), но широко используется технология CIDR. Благодаря этому, а также усовершенствованной системе групповой адресации и введению адресов нового типа IPv6 позволяет *снизить затраты на маршрутизацию*.

Произошли и чисто внешние изменения — разработчики стандарта предложили использовать вместо десятичной *шестнадцатеричную* форму записи IP-адреса. Каждые четыре шестнадцатеричные цифры отделяются друг от друга двоеточием. Вот как, например, может выглядеть адрес IPv6: FEDC:0A98:0:0:0:0:7654:3210. Для сетей, поддерживающих обе версии протокола (IPv4 и IPv6), разрешается задействовать для младших 4 байтов традиционную для IPv4 десятичную запись: 0:0:0:0:0:FFFF:129.144.52.38.

В новой версии IPv6 предусмотрено три основных типа адресов: индивидуальные адреса, групповые адреса и адреса произвольной рассылки. Мы уже обсуждали назначение этих типов адресов ранее. Тип адреса определяется значением нескольких старших битов адреса, которые названы **префиксом формата**. Индивидуальные адреса делятся на несколько подтипов.

Основным подтипом индивидуального адреса является **глобальный агрегируемый уникальный адрес**. Такие адреса могут агрегироваться для упрощения маршрутизации. В отличие от уникальных адресов узлов версии IPv4, которые состоят из двух полей — номера сети и номера узла, — глобальные агрегируемые уникальные адреса IPv6 имеют более сложную структуру, включающую шесть полей (рис. 18.20).

3	13	8	24	16	64
FP	TLA		NLA	SLA	Идентификатор интерфейса

Рис. 18.20. Структура глобального агрегируемого уникального адреса в пакете IPv6

- **Префикс формата** (Format Prefix, FP) для этого типа адресов имеет размер 3 бита и значение 001.
- **Поле TLA** (Top-Level Aggregation, TLA) предназначено для идентификации сетей самых крупных поставщиков услуг. Конкретное значение этого поля представляет собой общую часть адресов, которыми располагает данный поставщик услуг. Сравнительно небольшое количество разрядов, отведенных под это поле (13), выбрано специально для ограничения размера таблиц маршрутизации в магистральных маршрутизаторах

самого верхнего уровня Интернета. Это поле позволяет перенумеровать 8196 сетей поставщиков услуг верхнего уровня, а значит, число записей, описывающих маршруты между этими сетями, также будет ограничено значением 8196, что ускорит работу магистральных маршрутизаторов. Следующие 8 разрядов зарезервированы на будущее для расширения при необходимости поля TLA.

- **Поле NLA** (Next-Level Aggregation, NLA) предназначено для нумерации сетей средних и мелких поставщиков услуг. Значительный размер поля NLA позволяет путем агрегирования адресов отразить многоуровневую иерархию поставщиков услуг.
- **Поле SLA** (Site-Level Aggregation, SLA) предназначено для адресации подсетей отдельного абонента, например подсетей одной корпоративной сети.
- **Идентификатор интерфейса** является аналогом номера узла в IPv4. Отличием версии IPv6 является то, что в общем случае идентификатор интерфейса просто *совпадает с его локальным* (*аппаратным*) *адресом*, а не представляет собой произвольно назначенный администратором номер узла. Идентификатор интерфейса имеет длину 64 бита, что позволяет поместить туда MAC-адрес (48 бит), адрес конечного узла ATM (48 бит) или номер виртуального соединения ATM (до 28 бит), а также, вероятно, даст возможность использовать локальные адреса технологий, которые могут появиться в будущем. Такой подход *делает ненужным протокол ARP*, поскольку процедура отображения IP-адреса на локальный адрес становится тривиальной — она сводится к простому отбрасыванию старшей части адреса. Кроме того, в большинстве случаев *отпадает необходимость ручного конфигурирования* конечных узлов, так как младшую часть адреса — идентификатор интерфейса — узел узнает от аппаратуры (сетевого адаптера и т. п.), а старшую — номер подсети — ему сообщает маршрутизатор.

Рассмотрим пример (рис. 18.21). Пусть клиент получил от поставщика услуг пул адресов IPv6, определяемый префиксом 20:0A:00:C9:74:05/48. Поскольку первые три бита этого числа равны 001, это — *глобальный агрегируемый уникальный адрес*.

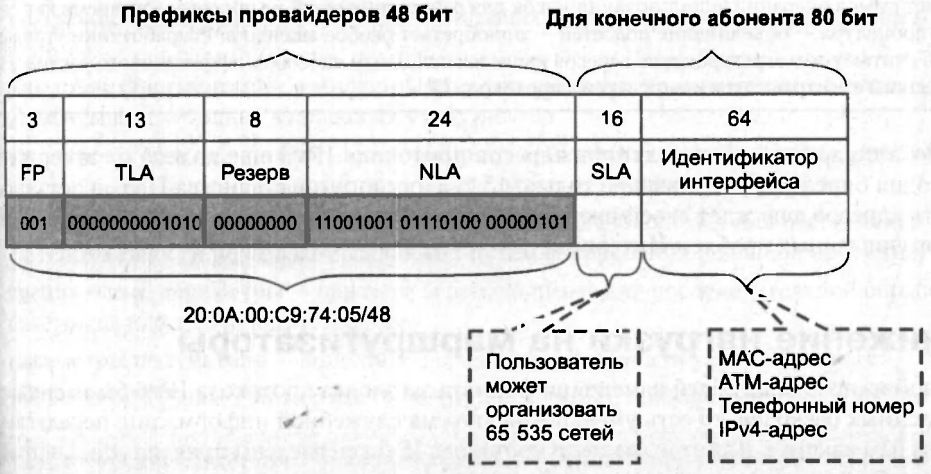


Рис. 18.21. Пример глобального агрегируемого адреса

Адрес этот принадлежит поставщику услуг верхнего уровня, у которого все сети имеют префикс 20:0A/16. Он может выделить поставщику услуг второго уровня некоторый

диапазон адресов с общим префиксом, образованным его собственным префиксом, а также частью поля NLA. Длина поля NLA, отводимая под префикс, определяется маской, которую поставщик услуг верхнего уровня также должен сообщить своему клиенту – поставщику услуг второго уровня. Пусть в данном примере маска состоит из 32 единиц в старших разрядах, а результирующий префикс поставщика услуг второго уровня имеет вид 20:0A:00:C9/32.

В распоряжении поставщика услуг второго уровня остается 16 разрядов поля NLA для нумерации сетей своих клиентов. В качестве клиентов могут выступать поставщики услуг третьего и более низких уровней, а также конечные абоненты – предприятия и организации. Пусть, например, следующий байт (01110100) в поле NLA поставщик услуг использовал для передачи поставщику услуг более низкого (третьего) уровня, а тот, в свою очередь, использовал последний байт поля NLA для назначения пула адресов клиенту. Таким образом, с участием поставщиков услуг трех уровней был сформирован префикс 20:0A:00:C9:74:05/48, который получил клиент.

Протокол IPv6 оставляет в полном распоряжении клиента 2 байта (поле SLA) для нумерации сетей и 8 байт (поле идентификатора интерфейса) для нумерации узлов. Имея такой огромный диапазон номеров подсетей, администратор получает широкие возможности. Для сравнительно небольшой сети он может выбрать плоскую организацию, назначая каждой имеющейся подсети произвольные неповторяющиеся значения из диапазона в 65 535 адресов, игнорируя оставшиеся. В крупных сетях более эффективным способом (сокращающим размеры таблиц корпоративных маршрутизаторов) может оказаться иерархическая структуризация сети на основе *агрегирования адресов*. В этом случае используется та же технология CIDR, но уже не поставщиком услуг, а администратором корпоративной сети.

ПРИМЕЧАНИЕ

Очевидно, что при таком изобилии сетей, которое предоставляется клиенту в IPv6, совершенно теряет смысл операция использования масок для разделения сетей на подсети, в то время как обратная процедура – объединение подсетей – приобретает особое значение. Разработчики стандартов IPv6 считают, что агрегирование адресов является основным способом эффективного расходования адресного пространства в новой версии протокола IP.

Работа по детализации подтипов адресов протокола IPv6 еще далека от завершения. Сегодня определено назначение только 15 % адресного пространства IPv6, а оставшаяся часть адресов еще ждет своей очереди, чтобы найти применение для решения одной из многочисленных проблем Интернета.

Снижение нагрузки на маршрутизаторы

Одной из основных целей изменения формата заголовка протокола IPv6 было снижение накладных расходов, то есть уменьшение объема служебной информации, передаваемой с каждым пакетом. Для этого в новом протоколе IP были введены понятия основного и дополнительных заголовков. Основной заголовок присутствует всегда, а необязательные дополнительные заголовки могут содержать, например, информацию о фрагментации исходного пакета, полный маршрут следования пакета при маршрутизации от источника, информацию, необходимую для защиты передаваемых данных.

Основной заголовок имеет фиксированную длину в 40 байт, его формат показан на рис. 18.22.

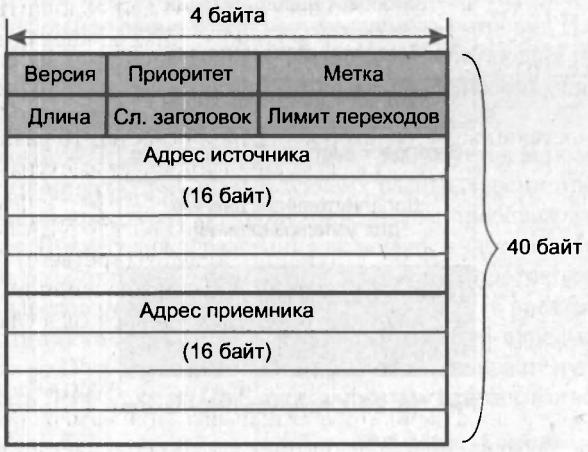


Рис. 18.22. Формат основного заголовка

Поле следующего заголовка соответствует по назначению полю протокола в версии IPv4 и содержит данные, определяющие тип заголовка, который следует за данным. Каждый следующий дополнительный заголовок также содержит поле следующего заголовка. Если IP-пакет не содержит дополнительных заголовков, то в этом поле будет значение, закрепленное за протоколом TCP, UDP, RIP, OSPF или другим, определенным в стандарте IPv4.

В предложениях по поводу протокола IPv6 фигурируют пока следующие типы дополнительных заголовков:

- **заголовок маршрутизации** — указание полного маршрута при маршрутизации от источника;
- **заголовок фрагментации** — информация, относящаяся к фрагментации IP-пакета (поле обрабатывается только в конечных узлах);
- **заголовок аутентификации** — информация, необходимая для аутентификации конечных узлов и обеспечения целостности содержимого IP-пакетов;
- **заголовок системы безопасности** — информация, необходимая для обеспечения конфиденциальности передаваемых данных путем шифрования и дешифрирования;
- **специальные параметры** — параметры необходимые для последовательной обработки пакетов на каждом маршрутизаторе;
- **параметры получателя** — дополнительная информация для узла назначения.

Таким образом, IP-пакет может иметь, например, формат, показанный на рис. 18.23.

Поскольку для маршрутизации пакета обязательным является лишь основной заголовок (почти все дополнительные заголовки обрабатываются только в конечных узлах), *это снижает нагрузку на маршрутизаторы*. В то же время возможность использования большого количества дополнительных параметров *расширяет функциональность протокола IP* и делает его открытым для внедрения новых механизмов.

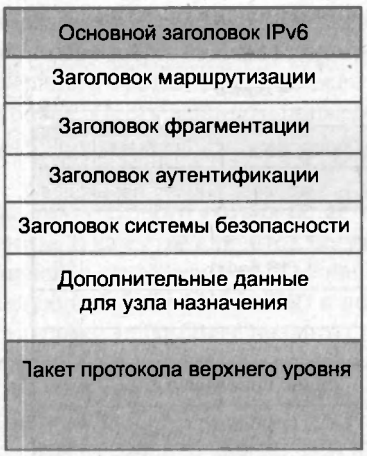


Рис. 18.23. Структура IPv6-пакета

Для того чтобы повысить производительность маршрутизаторов Интернета в части выполнения их основной функции — продвижения пакетов, в версии IPv6 предпринят ряд мер по освобождению маршрутизаторов от некоторых вспомогательных задач.

- *Перенесение функций фрагментации с маршрутизаторов на конечные узлы.* Конечные узлы в версии IPv6 обязаны найти минимальное значение MTU вдоль всего пути, соединяющего исходный узел с узлом назначения (эта техника под названием Path MTU Discovery уже используется в IPv4). Маршрутизаторы IPv6 не выполняют фрагментацию, а только посылают ICMP-сообщение о слишком длинном пакете конечному узлу, который должен уменьшить размер пакета.
- *Агрегирование адресов* ведет к уменьшению размера адресных таблиц маршрутизаторов, а значит, — к сокращению времени просмотра и обновления таблиц. При этом также сокращается служебный трафик, порождаемый протоколами маршрутизации.
- *Широкое использование маршрутизации от источника.* При маршрутизации от источника узел-источник задает полный маршрут прохождения пакета через сети. Такая техника освобождает маршрутизаторы от необходимости просмотра адресных таблиц при выборе следующего маршрутизатора.
- *Отказ от обработки не обязательных параметров заголовка.*
- *Использование в качестве номера узла его MAC-адреса* избавляет маршрутизаторы от необходимости применять протокол ARP.

Новая версия протокола IP, являющаяся составной частью проекта IPv6, предлагает встроенные средства защиты данных. Размещение средств защиты на сетевом уровне делает их прозрачными для приложений, так как между уровнем IP и приложением всегда будет работать протокол транспортного уровня. Приложения переписывать при этом не придется. Новая версия протокола IP со встроенными средствами обеспечения безопасности называется **IPSec** (Security Internet Protocol — защищенный протокол IP). Возможности этого протокола подробно рассматриваются в главе 24.

Переход на версию IPv6

При разработке IPv6 была предусмотрена возможность плавного перехода к новой версии, когда довольно значительное время будут сосуществовать островки Интернета, работающие по протоколу IPv6, и остальная часть Интернета, работающая по протоколу IPv4. Существует несколько подходов к организации взаимодействия узлов, использующих разные стеки TCP/IP.

- *Трансляция протоколов*. Трансляция протоколов реализуется шлюзами, которые устанавливаются на границах сетей, использующих разные версии протокола IP. Согласование двух версий протокола IP происходит путем преобразования пакетов IPv4 в IPv6, и наоборот. Процесс преобразования включает, в частности, отображение адресов сетей и узлов, различным образом трактуемых в этих протоколах. Для упрощения преобразования адресов между версиями разработчики IPv6 предлагают использовать специальный подтип IPv6-адреса — **IPv4-совместимый IPv6-адрес**, который в младших 4-х байтах переносит IPv4-адрес, а в старших 12 байтах содержит нули (рис. 18.24). Это позволяет получать IPv4-адрес из IPv6-адреса простым отбрасыванием старших байтов.

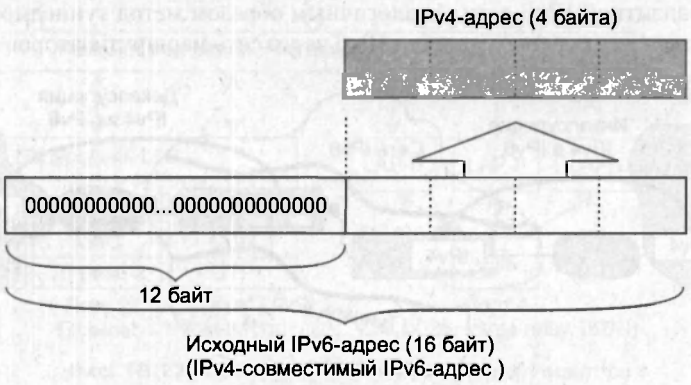


Рис. 18.24. Преобразование IPv6 в IPv4

Для решения обратной задачи — передачи пакетов IPv4 через части Интернета, работающие по протоколу IPv6, — предназначен **IPv4-отображенный IPv6-адрес**. Этот тип адреса также содержит в 4-х младших байтах IPv4-адрес, в старших 10-ти байтах — нули, а в 5-м и 6-м байтах IPv6-адреса — единицы, которые показывают, что узел поддерживает только версию 4 протокола IP (рис. 18.25).

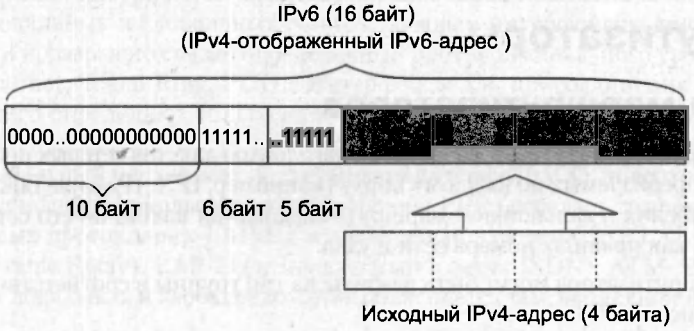


Рис. 18.25. Преобразование IPv4 в IPv6

- *Мультиплексирование стеков протоколов*. Мультиплексирование стеков протоколов означает установку на взаимодействующих хостах сети обеих версий протокола IP. Обе версии стека протоколов должны быть развернуты также на разделяющих эти хосты маршрутизаторах. В том случае, когда IPv6-хост отправляет сообщение IPv6-хосту, он использует стек IPv6, а если тот же хост взаимодействует с IPv4-хостом — стек IPv4. Маршрутизатор с установленными на нем двумя стеками называется маршрутизатором IPv4/IPv6, он способен обрабатывать трафики разных версий независимо друг от друга.
- *Инкапсуляция*, или *туннелирование*. Инкапсуляция — это еще один метод решения задачи согласования сетей, использующих разные версии протокола IP. Инкапсуляция может быть применена, когда две сети одной версии протокола, например IPv4, необходимо соединить через транзитную сеть, работающие по другой версии, например IPv6 (рис. 18.26) При этом пакеты IPv4 помещаются в пограничных устройствах (на рисунке роль согласующих устройств исполняют маршрутизаторы) в пакеты IPv6 и переносятся через «туннель», проложенный в IPv6-сети. Такой способ имеет недостаток, заключающийся в том, что узлы IPv4-сетей не имеют возможности взаимодействовать с узлами транзитной IPv6-сети. Аналогичным образом метод туннелирования может использоваться для переноса пакетов IPv6 через сеть маршрутизаторов IPv4.

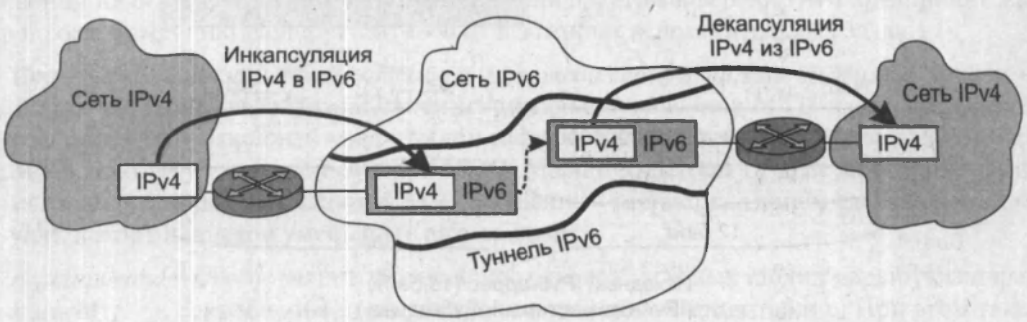


Рис. 18.26. Согласование технологий IPv4 и IPv6 путем туннелирования (инкапсуляции)

Переход от версии IPv4 к версии IPv6 только начинается. Сегодня уже существуют фрагменты Интернета, в которых маршрутизаторы поддерживают обе версии протокола. Эти фрагменты объединяются между собой через Интернет, образуя так называемую магистраль **6Bone**.

Маршрутизаторы

Функции маршрутизаторов

Основная функция маршрутизатора — чтение заголовков пакетов сетевых протоколов, принимаемых и буферизуемых по каждому порту (например, IPX, IP, AppleTalk или DECnet) и принятие решения о дальнейшем маршруте следования пакета по его сетевому адресу, включающему, как правило, номера сети и узла.

Функции маршрутизатора могут быть разбиты на три группы в соответствии с уровнями одели OSI (рис. 18.27).

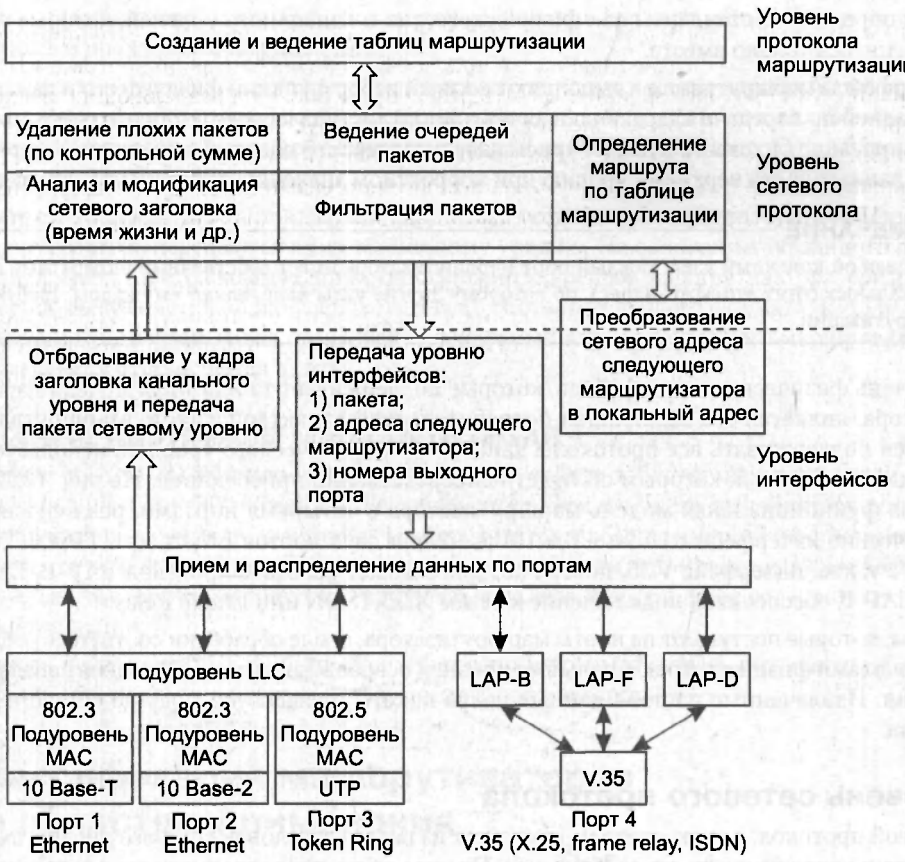


Рис. 18.27. Функциональная модель маршрутизатора

Уровень интерфейсов

На нижнем уровне маршрутизатор, как и любое устройство, подключенное к сети, обеспечивает физический интерфейс со средой передачи, включая согласование уровней электрических сигналов, линейное и логическое кодирование, оснащение определенным типом разъема. В разных моделях маршрутизаторов часто предусматриваются различные наборы физических интерфейсов, представляющих собой комбинацию портов для подсоединения локальных и глобальных сетей. С каждым интерфейсом для подключения локальной сети неразрывно связан определенный протокол канального уровня, например семейства Ethernet, Token Ring, FDDI. Интерфейсы для присоединения к глобальным сетям чаще всего определяют только некоторый стандарт физического уровня, поверх которого в маршрутизаторе могут работать различные протоколы канального уровня. Например, глобальный порт может поддерживать интерфейс V.35, поверх которого могут работать различные протоколы канального уровня: PPP (передает трафик протокола IP и других сетевых протоколов), LAP-B (используемый в сетях X.25), LAP-F (используемый в сетях Frame Relay), LAP-D (используемый в сетях ISDN), ATM. Разница между интерфейсами локальных и глобальных сетей объясняется тем, что технологии локальных

сетей определяют стандарты как физического, так и канального уровней, которые могут применяться только вместе.

Интерфейсы маршрутизатора выполняют полный набор функций физического и канального уровней по передаче кадра, включая получение доступа к среде (если это необходимо), формирование битовых сигналов, прием кадра, подсчет его контрольной суммы и передачу поля данных кадра верхнему уровню при корректном значении контрольной суммы.

ПРИМЕЧАНИЕ

Как и любой конечный узел, каждый порт маршрутизатора имеет собственный аппаратный адрес (в локальных сетях это MAC-адрес), по которому другие узлы направляют ему кадры, требующие маршрутизации.

Перечень физических интерфейсов, которые поддерживает та или иная модель маршрутизатора, является его важнейшей потребительской характеристикой. Маршрутизатор должен поддерживать все протоколы канального и физического уровней, используемые в каждой из сетей, к которым он будет непосредственно присоединен. На рис. 18.27 показана функциональная модель маршрутизатора с четырьмя портами, реализующими физические интерфейсы 10Base-T и 10Base-2 для двух портов Ethernet, UTP для Token Ring, а также интерфейс V.35, поверх которого может работать протокол LAP-B, LAP-D или LAP-F, обеспечивая подключение к сетям X.25, ISDN или Frame Relay.

Кадры, которые поступают на порты маршрутизатора, после обработки соответствующими протоколами физического и канального уровней освобождаются от заголовков канального уровня. Извлеченные из поля данных кадра пакеты передаются модулю сетевого протокола.

Уровень сетевого протокола

Сетевой протокол, в свою очередь, извлекает из пакета заголовок сетевого уровня, *анализирует и корректирует его содержимое*. Прежде всего проверяется контрольная сумма, и если пакет пришел поврежденным, он отбрасывается. Кроме того, выполняется проверка на превышение времени жизни пакета (время, которое пакет провел в сети). Если превышение имело место, то пакет также отбрасывается. На этом этапе вносятся корректировки в содержимое некоторых полей, например наращивается время жизни пакета, пересчитывается контрольная сумма.

На сетевом уровне выполняется одна из важнейших функций маршрутизатора — *фильтрация трафика*. Пакет сетевого уровня, находящийся в поле данных кадра, для мостов/коммутаторов представляется неструктурированной двоичной последовательностью. Маршрутизаторы же, программное обеспечение которых содержит модуль сетевого протокола, способны производить анализ *отдельных полей пакета*. Они оснащаются развитыми средствами пользовательского интерфейса, которые позволяют администратору без особых усилий задавать сложные правила фильтрации. Маршрутизаторы, как правило, позволяют также анализировать структуру сообщений транспортного уровня, поэтому фильтры могут не пропускать в сеть сообщений определенных прикладных служб, например службы telnet, анализируя поле типа протокола в транспортном сообщении.

Однако основной функцией сетевого уровня маршрутизатора является *определение маршрута пакета*. По номеру сети, извлеченному из заголовка пакета, модуль сетевого протокола находит в таблице маршрутизации строку, содержащую сетевой адрес следую-

щего маршрутизатора и номер порта, на который нужно передать данный пакет, чтобы он двигался в правильном направлении.

Перед тем как передать сетевой адрес следующего маршрутизатора на канальный уровень, необходимо преобразовать его в локальный адрес той технологии, которая используется в сети, содержащей следующий маршрутизатор. Для этого сетевой протокол обращается к *протоколу разрешения адресов*.

С сетевого уровня пакет, локальный адрес следующего маршрутизатора и номер порта маршрутизатора передаются вниз, канальному уровню. На основании указанного номера порта осуществляется коммутация с одним из интерфейсов маршрутизатора, средствами которого выполняется упаковка пакета в кадр соответствующего формата. В поле адреса назначения заголовка кадра помещается локальный адрес следующего маршрутизатора. Готовый кадр отправляется в сеть.

Уровень протокола маршрутизации

Сетевые протоколы активно используют в своей работе таблицу маршрутизации, но ни ее построением, ни поддержанием не занимаются. Эти функции выполняют протоколы маршрутизации, с помощью которых маршрутизаторы обмениваются информацией о топологии сети, а затем анализируют полученные сведения, определяя наилучшие по тем или иным критериям маршруты. Результаты анализа и составляют содержимое таблиц маршрутизации.

Помимо перечисленных функций на маршрутизаторы могут быть возложены и другие обязанности, например операции, связанные с фрагментацией.

Классификация маршрутизаторов по областям применения

По областям применения маршрутизаторы делятся на несколько классов (рис. 18.28).

Магистральные маршрутизаторы предназначены для построения магистральной сети оператора связи или крупной корпорации. Магистральные маршрутизаторы оперируют агрегированными информационными потоками, переносящими данные большого количества пользовательских соединений.

Для решения этой задачи магистральные маршрутизаторы оснащаются высокоскоростными интерфейсами, такими как ATM 155/622 Мбит/с, Gigabit Ethernet и 10G Ethernet, а также интерфейсами SONET/SDH со скоростями от 155 Мбит/с до 10 Гбит/с. Для получения отказоустойчивой топологии магистральной сети магистральные маршрутизаторы должны поддерживать несколько таких интерфейсов.

Очевидно, что для того чтобы не создавать «узких мест» в магистральной сети, магистральный маршрутизатор должен обладать очень высокой производительностью. Например, если маршрутизатор оснащен 8 интерфейсами по 10 Гбит/с (Ethernet или SDH), то его общая производительность должна составлять 80 Гбит/с. Для достижения такой производительности магистральные маршрутизаторы обладают распределенной внутренней архитектурой, подобной архитектуре коммутаторов локальных сетей. Каждый порт или группа портов оснащается *собственным процессором*, который самостоятельно выполняет

продвижение IP-пакетов на основании локальной копии таблицы маршрутизации. Для передачи пакетов между портами служит *коммутирующий блок* на основе разделяемой памяти, общей шины или коммутатора каналов. Общие задачи, включая построение таблицы маршрутизации, хранение конфигурационных параметров, удаленное управление маршрутизатором и т. п., решает *центральный блок управления*.

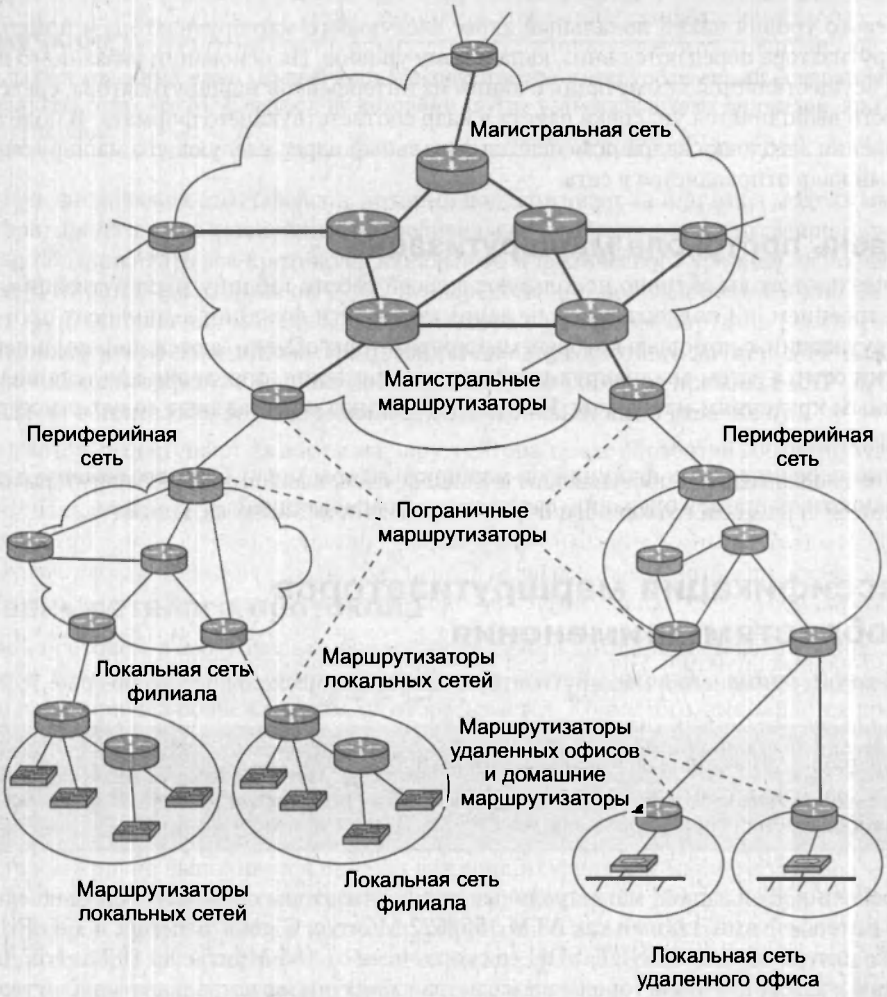


Рис. 18.28. Классы маршрутизаторов

Понятно, что функции продвижения IP-пакетов существенно сложнее, чем продвижения кадров Ethernet и других технологий локальных сетей. Поэтому процессоры портов обычно не нагружают дополнительными функциями, такими как фильтрация трафика или трансляция адресов. Даже обеспечение параметров QoS не всегда реализуется таким процессором в полном объеме — обычно дело ограничивается поддержанием очередей, а до

профилирования трафика не доходит. Это связано с тем, что магистральный маршрутизатор работает внутри сети и не взаимодействует с внешним миром, а значит, не выполняет пограничные функции, требующие фильтрации и профилирования. Другими словами, основная задача магистрального маршрутизатора — передача пакетов между своими интерфейсами с как можно большей скоростью.

Большое количество интерфейсов, характерное для магистрального маршрутизатора, позволяет строить избыточные топологии, приближающиеся к полносвязной схеме, и тем самым обеспечивать отказоустойчивость сети. Однако и сам магистральный маршрутизатор должен обладать высокой надежностью. Надежность и отказоустойчивость маршрутизатора достигается за счет избыточных модулей, таких как центральные процессоры, процессоры портов, источники питания.

Пограничные маршрутизаторы, называемые также **маршрутизаторами доступа**, соединяют магистральную сеть с периферийными сетями. Эти маршрутизаторы образуют особый слой, который выполняет функции приема трафика от внешних по отношению к магистрали сетей.

Периферийная сеть часто находится под автономным административным управлением. Это может быть сеть клиента оператора связи, непосредственно присоединенная к его магистрали, или же сеть регионального отделения крупной корпорации, обладающей собственной магистралью.

В любом случае трафик, поступающий на интерфейсы пограничного маршрутизатора от сети, которую администратор магистрали не может контролировать, нужно фильтровать и профилировать. Поэтому к пограничному маршрутизатору предъявляются другие требования, нежели к магистральному. На первый план выступают его способности к *максимальной гибкости при фильтрации и профилировании трафика*. Кроме того, очень важно, чтобы производительность пограничного маршрутизатора не снижалась при выполнении этих дополнительных функций. Интерфейсы пограничного маршрутизатора менее скоростные, чем магистрального, но более разнообразные, так как ему приходится присоединять к магистрали сети различных технологий.

Деление маршрутизаторов на магистральные и пограничные не является строгим и четким. Такое деление просто отражает предпочтительную область применения маршрутизатора, где в наибольшей степени проявляются его преимущества. В то же время любой маршрутизатор можно применять не только в его профильной области. Так, магистральный маршрутизатор, оснащенный низкоскоростными портами, может одновременно играть роль пограничного. А маршрутизатор, хорошо исполняющий роль пограничного для крупной сети, может быть магистральным маршрутизатором для сети меньшего масштаба, где его интерфейсы вполне справятся с нагрузкой на магистраль.

Деление маршрутизаторов на магистральные и пограничные отражает только один аспект их применения, а именно их положение относительно собственной и внешних сетей. Понятно, что существуют и другие аспекты. Так, маршрутизаторы можно разделить на **маршрутизаторы операторов связи** и **корпоративные маршрутизаторы**.

Основным отличием корпоративных маршрутизаторов является их *высокая надежность*, а также *поддержка полного набора функций*, необходимых для коммерческой работы в Интернете, начиная от протокола BGP и кончая системами регистрации пользовательских

потоков данных, что необходимо для биллинговых схем. Необходимость высокой надежности объясняется значительной стоимостью простоя маршрутизатора при оказании коммерческих услуг. Требования к надежности услуг передачи данных постоянно растут, пользователи Интернета и виртуальных частных сетей хотят, чтобы эти услуги были такими же надежными, как услуги телефонной сети. Поэтому когда мы говорим о том, что готовность некоторых моделей маршрутизаторов достигла рубежа 0,999 и стремится к показателям телефонного оборудования в 0,99999, то в первую очередь это относится к маршрутизаторам операторов связи, как магистральным, так и пограничным. Корпоративные маршрутизаторы предназначены для применения в пределах корпоративной сети, поэтому требования к надежности здесь ниже, а функциональность для работы в Интернете в качестве самостоятельной автономной системы не требуется.

Конечно, характеристики маршрутизаторов операторов связи и корпоративных маршрутизаторов в значительной степени зависят от масштаба и специфики оператора связи или корпорации. Для крупного международного оператора связи сегодня требуются магистральные маршрутизаторы с интерфейсами 10 Гбит/с, которые в недалеком будущем будут заменены маршрутизаторами с портами 100 Гбит/с. Пограничные маршрутизаторы такого оператора также будут относиться к лучшим маршрутизаторам этого класса по производительности, работая с портами доступа со скоростями от 622 Мбит/с до 2,5 Гбит/с.

Менее крупным операторам связи, то есть региональным и локальным, такие высокопроизводительные маршрутизаторы не требуются, так как объемы передаваемого ими трафика гораздо меньше. Поэтому магистральный маршрутизатор подобного оператора может ограничиться поддержкой интерфейсов 1 Гбит/с, а пограничный маршрутизатор должен, кроме того, обеспечивать коммутируемый доступ абонентов через телефонные сети. В небольших сетях магистральных маршрутизаторов может не быть вообще, такая сеть будет состоять из нескольких (или даже одного) пограничных маршрутизаторов.

Аналогичная картина наблюдается и в корпоративных сетях, где также применяются маршрутизаторы различной производительности и надежности. Например, крупные корпорации могут применять магистральные и пограничные маршрутизаторы, близкие по характеристикам к маршрутизаторам операторов связи категории Tiar 1. Однако более обычной является ситуация, когда в корпоративных сетях применяется оборудование с характеристиками на один уровень ниже. Это значит, что крупные многонациональные корпорации задействуют оборудование, которое обычно используется региональными операторами и т. д., по нисходящей.

Маршрутизаторы региональных отделений соединяют региональные отделения между собой и с магистральной сетью. Сеть регионального отделения, так же как и магистральная сеть, может состоять из нескольких локальных сетей. Такой маршрутизатор обычно представляет собой некоторую упрощенную версию магистрального корпоративного маршрутизатора.

Если он выполнен на основе шасси, то количество слотов его шасси меньше (4–5). Возможен также конструктив с фиксированным количеством портов. Поддерживаемые интерфейсы локальных и глобальных сетей менее скоростные. Это наиболее обширный класс выпускаемых маршрутизаторов, характеристики которых могут приближаться к характеристикам магистральных маршрутизаторов, а могут и опускаться до характеристик маршрутизаторов удаленных офисов.

Маршрутизаторы удаленных офисов соединяют, как правило, единственную локальную сеть удаленного офиса с магистральной сетью или сетью регионального отделения по глобальной связи.

Как правило, интерфейс локальной сети представляет собой Ethernet 100/1000 Мбит/с, а интерфейс глобальной сети — выделенную линию со скоростью 2–100 Мбит/с. Маршрутизатор удаленного офиса может поддерживать работу по коммутируемой телефонной линии в качестве резервной связи для выделенного канала. Существует очень большое количество типов маршрутизаторов удаленных офисов. Это объясняется как массовостью потенциальных потребителей, так и специализацией такого типа устройств, проявляющейся в поддержке какого-либо конкретного типа глобальной связи. Например, существуют маршрутизаторы, работающие только в сетях ISDN, существуют модели только для аналоговых выделенных линий и т. п.

Чем меньше требований предъявляется к производительности маршрутизатора, тем более вероятно, что он выполнен по классической схеме первых маршрутизаторов (и мостов локальных сетей), то есть схемы на основе единственного центрального процессора и без процессоров портов. Такая схема гораздо дешевле, но ее производительность полностью определяется производительностью процессора и не масштабируется с ростом числа портов.

Программный маршрутизатор, являясь одной из популярных реализаций такой схемы, представляет собой программный модуль универсальной операционной системы семейства Unix или Windows.

И только появление в глобальных сетях высокоскоростных технологий, таких как ATM, Ethernet, SONET/SDH, DWDM, привело к резкому повышению требований к производительности маршрутизаторов, в результате представители наиболее совершенного класса маршрутизаторов повсеместно перешли на *многопроцессорные схемы с коммутирующим блоком*, успешно опробованные на коммутаторах локальных сетей.

Маршрутизаторы локальных сетей предназначены для разделения крупных локальных сетей на подсети. Это особый класс маршрутизаторов, которые, как правило, не имеют интерфейсов глобальных сетей.

Многие маршрутизаторы этого типа ведут свое происхождение от коммутаторов локальных сетей, что и дало им второе название — **коммутаторы 3-го уровня**. Коммутаторы 3-го уровня выполняют все функции маршрутизаторов, но, кроме того, могут работать как обычные коммутаторы локальных сетей, то есть коммутаторы 2-го уровня. Режим работы (маршрутизатор или коммутатор) зависит от конфигурационных параметров. Возможен также комбинированный режим работы, когда несколько портов коммутатора 3-го уровня имеют один и тот же IP-адрес сети (рис. 18.29). В этом случае передача пакетов между группой портов, принадлежащих одной сети, выполняется в режиме коммутации на канальном уровне, то есть на основе MAC-адресов. Если же порты принадлежат разным IP-сетям, то тогда коммутатор выполняет маршрутизацию между сетями. Выбор режима передачи пакета определяется конфигурированием IP-адресов портов и, соответственно, компьютеров.

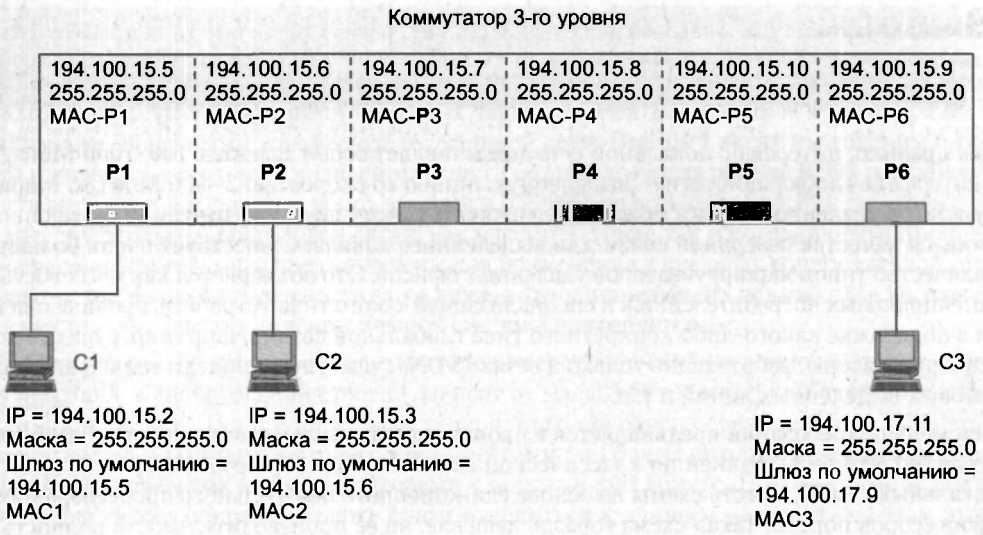


Рис. 18.29. Комбинированный режим работы коммутатора 3-го уровня

ПРИМЕР

Например, если два компьютера (C1 и C2 на рис. 18.29) имеют адреса, принадлежащие одной сети, то при обмене информацией они не будут передавать пакеты маршрутизатору по умолчанию, а задействуют протокол ARP, чтобы узнать MAC-адрес компьютера назначения. Пусть компьютеру C1 требуется передать пакет компьютеру C2. Коммутатор 3-го уровня передает кадр ARP-запроса компьютера C1 с широковещательным MAC-адресом всем портам, принадлежащим одной IP-сети, то есть портам P1, P2, P3 и P4. Компьютер C2 распознает свой IP-адрес (194.100.15.3) в этом запросе и отвечает направленным кадром с MAC-адресом назначения компьютера C1 (MAC1), помещая в ответ собственный MAC-адрес (MAC2). После этого компьютер C1 направляет IP-пакет компьютеру C2, помещая его в кадр с адресом назначения MAC2. Коммутатор 3-го уровня передает этот кадр с порта P1 на порт P2 в соответствии с алгоритмом моста на основе таблицы продвижения 2-го уровня. Аналогичным образом будет работать коммутатор 3-го уровня. В случае когда компьютеры принадлежат разным IP-сетям, поведение компьютера-отправителя диктует коммутатору 3-го уровня способ продвижения пакета. Если, например, компьютер C1 отправляет пакет компьютеру C3, находящемуся в другой сети, то он обязан передать пакет маршрутизатору по умолчанию, а не пытаться с помощью ARP узнать MAC-адрес компьютера назначения. Поэтому компьютер C1 делает ARP-запрос о MAC-адресе известного ему маршрутизатора по умолчанию, которым для него является порт P1 с IP-адресом IP-R1. После получения MAC-адреса порта P1 (MAC-P1) компьютер C1 посылает ему IP-пакет для компьютера C3 (то есть по IP-адресу назначения 194.100.17.11), оформив его как кадр Ethernet с адресом назначения MAC-P1. Получив кадр с собственным MAC-адресом, коммутатор 3-го уровня обрабатывает его по схеме маршрутизации, а не коммутации.

Коммутаторы 3-го уровня поддерживают технику VLAN, являясь основным типом устройств для соединения отдельных виртуальных сетей в составную IP-сеть. Обычно каждой виртуальной сети присваивается номер IP-сети, так что передача внутри сетей идет на основе MAC-адресов, а между сетями — на основе IP-адресов. В представленном на рис. 18.29 примере сети порты P1–P4 могут принадлежать одной виртуальной сети, а порты P5, P6 — другой.

Выводы

IP-маршрутизаторы позволяют фильтровать пользовательский трафик на основе различных признаков, включающих адреса источника и назначения, тип протокола, который переносят IP-пакеты, номера UPD- и TCP-портов и некоторые другие. Это свойство маршрутизаторов широко применяется для защиты сетей от атак злоумышленников и ограничения доступа легальных пользователей.

Фильтрация маршрутных объявлений обеспечивает управление связностью сетей в целом, предотвращая появление записей об определенных сетях в таблицах маршрутизации.

IP-маршрутизаторы уже долгое время поддерживают многие механизмы QoS: приоритетные и взвешенные очереди, профилирование трафика, обратную связь для TCP-трафика. Однако только в середине 90-х годов, когда через Интернет стал передаваться чувствительный к задержкам трафик, начались работы по созданию системы стандартов QoS для IP-сетей.

Сегодня существует две системы стандартов QoS для IP-сетей — IntServ и DiffServ. Первая обеспечивает гарантированное качество обслуживания микропотоков, используя сигнальный протокол RSVP для резервирования ресурсов маршрутизаторов. Недостатком такого подхода является большая нагрузка на магистральные маршрутизаторы, которые должны хранить информацию о состоянии тысяч пользовательских потоков.

В технологии DiffServ используется агрегированный подход, когда качество обслуживания обеспечивается для небольшого количества классов трафика. Это существенно снижает нагрузку на маршрутизаторы.

Типичный маршрутизатор представляет собой программируемое вычислительное устройство, которое работает под управлением специализированной операционной системы, оптимизированной для выполнения операций построения таблиц маршрутизации и продвижения пакетов на их основе.

Маршрутизатор часто строится по мультипроцессорной схеме, причем используется симметричное мультипроцессирование, асимметричное мультипроцессирование и их сочетание.

Маршрутизаторы можно классифицировать различными способами. Их можно разделить на магистральные и пограничные (по положению относительно границ сети), на маршрутизаторы операторов связи и корпоративные маршрутизаторы (в зависимости от типа предприятия, владеющего сетью).

Технология трансляции сетевых адресов (NAT) позволяет предприятию решить проблему дефицита IP-адресов, а также повысить безопасность сети путем сокрытия адресов узлов своей сети за счет использования во внутренней сети частных адресов, которые при выходе пакета во внешнюю сеть транслируются в глобальные IP-адреса.

Вопросы и задания

1. Чем результат фильтрации объявлений маршрутизации отличается от результата фильтрации пользовательского трафика?
2. Какую смысловую нагрузку несет термин «интегрированные» в названии технологии IntServ?
3. Какой параметр можно использовать, чтобы ограничить пульсацию входного потока пакетов, профилируемого по алгоритму ведра маркеров? Варианты ответов:

   а) объем маркера;

   б) скорость наполнения ведра;

   в) интервал поступления маркеров;

   г) объем ведра маркеров.
4. Почему для UDP-трафика неприменим механизм RED?

5. В чем назначение технологии NAT? Варианты ответов:
   а) отражение DOS-атак;
   б) решение проблемы дефицита адресов в протоколе IPv4;
   в) защита внутреннего адресного пространства сети предприятия.
6. Заполните столбец «Назначенный порт» в таблице.

Частный адрес	Порт отправителя	Глобальный адрес	Назначенный порт
10.0.25.1	1035	193.55.13.79	
10.0.25.2	1035	193.55.13.79	
10.0.25.2	1047	193.55.13.79	

7. Протокол IGMP используется при взаимодействии:
   а) маршрутизатора с получателем группового трафика;
   б) источника группового трафика с маршрутизатором;
   в) маршрутизаторов, передающих групповой трафик.
8. В чем состоит принципиальное отличие протоколов маршрутизации группового вещания плотного режима от соответствующих протоколов разряженного режима?
9. Каково отношение администратора IPv6-сети к маскам? Варианты ответов:
   а) использует и для объединения подсетей, и для разделения на подсети;
   б) использует для разделения на подсети;
   в) использует для объединения подсетей;
   г) игнорирует как ненужное средство.

Часть V

Технологии глобальных сетей

Технология IP, которую мы рассматривали в предыдущей части книги, позволяет строить составные сети различного типа, как локальные, так и глобальные. Протокол IP является сегодня тем протоколом, который объединяет многочисленные сети операторов связи и предприятий в глобальную мировую компьютерную сеть, называемую Интернетом.

Превращение Интернета в мировую компьютерную сеть привело к тому, что одной из основных услуг операторов связи, относящейся к транспортным услугам компьютерных сетей, стал доступ в Интернет, а операторы связи по совместительству стали поставщиками, или провайдерами, услуг Интернета. Другим популярным типом услуг сетей операторов связи является услуга виртуальных частных сетей, которая позволяет объединить отдельные территориально рассредоточенные сети некоторого предприятия в единую корпоративную сеть.

Технология IP не является единственной технологией коммутации пакетов, которая работает в глобальных сетях. Типичная глобальная сеть имеет многоуровневую структуру, в которой IP занимает верхний уровень (если рассматривать только уровни, обеспечивающие транспорт), а под уровнем IP работают пакетные технологии канального уровня. Для глобальных сетей был разработан ряд технологий, учитывающих особенности этого типа сетей, в частности X.25 (она сегодня представляет только исторический интерес), Frame Relay (FR) и ATM. Объединяет все перечисленные технологии то, что они основаны на технике виртуальных каналов. Основная причина успеха техники виртуальных каналов в глобальных сетях состоит в том, что она обеспечивает гораздо более высокую степень контроля над соединениями между пользователями сети и путями прохождения информационных потоков через узлы сети, чем дейтаграммная техника.

После прихода IP в сети операторов связи дейтаграммная техника и техника виртуальных каналов стали дополнять друг друга, и во многих случаях протокол IP работает поверх Frame Relay или ATM. Тем самым пользователю предоставляются услуги IP, а трафик пользователи внутри сети провайдера переносится по виртуальным каналам, что делает эту операцию более надежной и контролируемой.

Помимо поддержания трафика протокола IP внутри сети оператора связи, технологии Framne Relay и ATM долгое время давали операторам связи возможность предоставлять пользователям услуги виртуальных частных сетей; при этом виртуальные каналы Frane Relay или ATM прозрачным образом соединяли сети предприятия. В главе 19 рассматриваются общие принципы организации глобальных сетей и их услуг, а также дается обзор технологий Frame Relay, ATM и особенностей работы IP в глобальных сетях.

Опыт сосуществования IP с технологиями, основанными на механизме виртуальных каналов, привел в середине 90-х годов к появлению гибридной технологии MPLS, которая тесно интегрирована

с протоколами стека IP, так что иногда ее называют IP/MPLS. При использовании MPLS протоколы маршрутизации стека TCP/IP служат для исследования топологии сети и нахождения рациональных маршрутов, а продвигаются пакеты на основе техники виртуальных каналов. Интеграция IP и MPLS оказалось очень удачной, так что эта комбинация в настоящее время почти вытеснила из глобальных сетей технологии Frame Relay и ATM, переведя их в статус унаследованных технологий, то есть таких, которые все еще работают, но перспективы развития уже не имеют.

MPLS сегодня используется в различных качествах, и как внутренняя технология операторов связи, дающая высокую степень контроля над трафиком и обеспечивающая быстрое восстановление соединений, и как технология, на которой строятся услуги оператора связи, в первую очередь — услуга виртуальных частных сетей. Технология MPLS и ее основные приложения рассматриваются в главе 20.

Сравнительно недавно класс технологий глобальных сетей пополнился новым представителем — Ethernet операторского класса (Carrier Ethernet). Этим именем одновременно называют и услугу виртуальных частных сетей, которая предоставляется пользователем с интерфейсом Ethernet, и усовершенствованную версию классической технологии Ethernet, снабженную некоторыми новыми свойствами, необходимыми для успешной работы в глобальных сетях. Глава 21 посвящена описанию Ethernet как услуги глобальных сетей, кроме того, в этой главе рассматриваются способы реализации этой услуги на основе усовершенствований, внесенных в традиционную технологию Ethernet, и на основе использования в сети оператора связи технологии MPLS (последний вариант также известен под названием VPLS).

Обеспечение высокоскоростного доступа к сетевой магистрали представляет собой сегодня масштабную и специфическую проблему. Действительно, скорость нужно повысить на миллионах линий связи, соединяющих помещения пользователей с ближайшими центральными офисами операторов связи. Поэтому традиционные для магистрали решения, основанные на применении оптического волокна и требующие прокладки новых кабелей к домам и офисным зданиям, для обеспечения массового доступа часто оказываются экономически не оправданными. Более эффективными являются технологии, в которых задействуется существующая кабельная инфраструктура (например, линии ADSL, работающие на абонентских окончаниях телефонной сети) или кабельные модемы, использующие системы кабельного телевидения. Альтернативным решением является беспроводной доступ, причем как мобильный, так и фиксированный. Схемы и технологии доступа рассматриваются в главе 22.

Глобальные сети предоставляют не только транспортные услуги. Интернет стал популярным в первую очередь благодаря своим информационным сервисам, таким как электронная почта и WWW. Растет популярность и новых сервисов Интернета, в первую очередь это IP-телефония и сервис видеоконференций; совсем недавно началось и телевещание через Интернет (IPTV). Прикладные сервисы глобальных сетей рассматриваются в главе 23.

Часть, а вместе с ней и книга, завершается главой 24, которая посвящена обеспечению безопасности транспортной системы сети. Уязвимость Интернета является оборотной стороной его открытости, так как в Интернете каждый может не только общаться с каждым, но и атаковать каждого. Вирусы, черви, распределенные атаки и, наконец, спам — все это, к сожалению, ежедневно мешает «жителям» Интернета нормально жить и работать. В главе 24 анализируются основные типы угроз, присущих глобальным сетям, и изучаются базовые механизмы и технологии защиты от этих угроз.

- Глава 19. Транспортные услуги и технологии глобальных сетей
- Глава 20. Технология MPLS
- Глава 21. Ethernet операторского класса
- Глава 22. Удаленный доступ
- Глава 23. Сетевые службы
- Глава 24. Сетевая безопасность

ГЛАВА 19 Транспортные услуги и технологии глобальных сетей

Транспортные технологии и услуги глобальных сетей являются тем фундаментом, на котором строятся технологии и услуги прикладного уровня, такие как электронная почта или WWW.

Базовые понятия

Типы публичных услуг сетей операторов связи

Сегодня существует единственная мировая глобальная компьютерная сеть — Интернет. Основу Интернета составляют компьютерные сети операторов связи, которые предоставляют своим клиентам — предприятиям и индивидуальным пользователям — разнообразные услуги, в том числе транспортные, с помощью которых клиенты объединяют свои локальные сети в глобальные. Благодаря такому особому положению требования операторов связи к технологиям глобальных компьютерных сетей являются решающими при их разработке. Поэтому перед рассмотрением конкретных технологий глобальных компьютерных сетей полезно исследовать основные типы транспортных услуг операторов связи, так как специфика этих услуг и определяет специфику технологий. Поскольку сеть оператора связи служит для предоставления не только услуг компьютерных сетей, но и традиционных услуг телефонии, последние также оказывают влияние на структуру сети и применяемые в ней технологии.

Выделенные каналы для построения частной сети

В течение довольно длительного начального периода своего существования (до интернет-революции, то есть до начала 90-х годов) корпоративные компьютерные сети представляли собой частные сети. Это значит, что сеть предприятия была полностью или почти полностью изолирована от сетей других предприятий, при этом все локальные сети предприятия, расположенные в разных городах (эти сети часто называют сайтами, подчеркивая их территориальную рассредоточенность), соединялись физическими каналами только между собой. Такие физические каналы либо принадлежали самому предприятию (довольно дорогой и поэтому редко встречавшийся вариант), либо брались в аренду у операторов связи и назывались **выделенными**, или **арендуемыми**, **каналами**. Первое название подчеркивает тот факт, что канал постоянно коммутируется так, что вся его фиксированная пропускная способность выделяется клиенту. В начальный период создания глобальных компьютерных сетей выделенные линии представляли собой постоянно скоммутированные аналоговые телефонные соединения.

По мере роста популярности компьютерных сетей услуги выделенных каналов стали более востребованными, и такой сервис стали предоставлять в более широком масштабе на основе новых технологий первичных сетей: PDH, SDH, OTN и DWDM.

На рис. 19.1 показан пример построения корпоративной сети клиента *A* с помощью сервиса выделенных каналов. Сети 2 и 3 этого клиента соединены двумя выделенными каналами с сетью 1 того же клиента, образуя корпоративную сеть со звездообразной топологией. Выделенные каналы проложены через сети операторов 1 и 2.

Виртуальная частная сеть

Сервис **виртуальных частных сетей** (Virtual Private Network, VPN) появился как более экономичная альтернатива сервису выделенных каналов. Каналы виртуальной частной сети, так же как и выделенные каналы, соединяют отдельные сети клиента этой услуги в единую изолированную сеть. Однако в отличие от выделенных каналов, которые строятся с помощью техники коммутации каналов и поэтому обладают фиксированной

пропускной способностью, реально выделенной данному клиенту, каналы виртуальной частной сети проложены внутри сети с коммутацией пакетов, такой как IP, Frame Relay или Ethernet.

На рис. 19.2 показан тот же пример, что и на рис. 19.1, но в данном случае корпоративная сеть клиента *A* построена с помощью сервиса виртуальной частной сети, и каналы представляют собой соединения в сетях с коммутацией пакетов операторов 1 и 2.

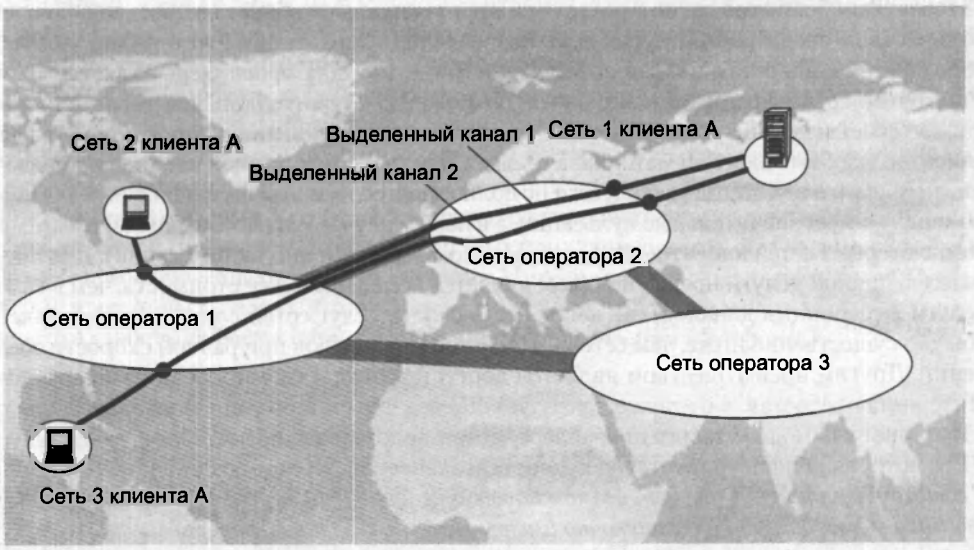


Рис. 19.1. Сервис выделенных каналов

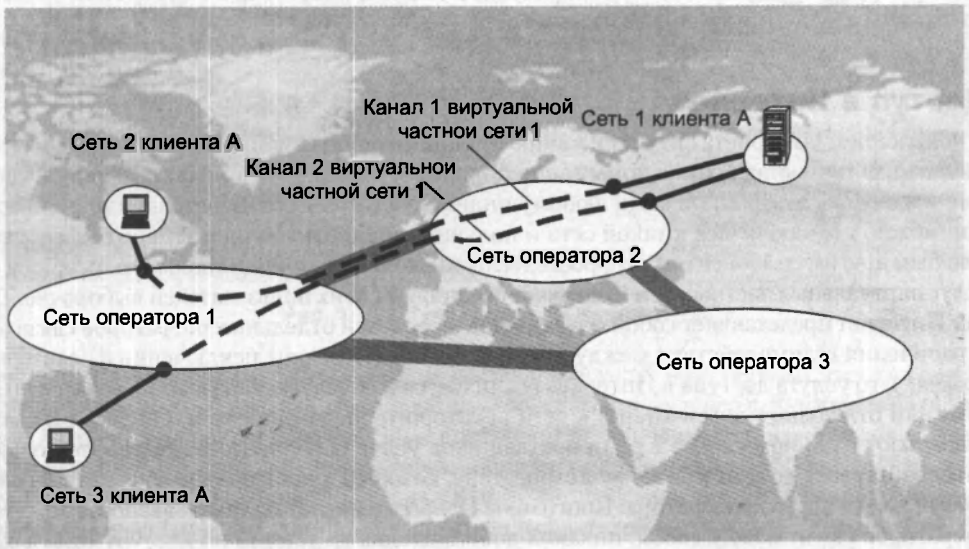


Рис. 19.2. Сервис виртуальной частной сети

Технология VPN позволяет с помощью разделяемой несколькими предприятиями сетевой инфраструктуры реализовать сервисы, приближающиеся к сервисам частной сети по качеству (безопасность, доступность, предсказуемая пропускная способность, независимость в выборе адресов), но на разделяемой между пользователями инфраструктуре публичной сети с коммутацией пакетов, такой как Frame Relay или IP.

Так как каналы виртуальной частной сети являются соединениями в публичной сети с коммутацией пакетов, то они разделяют пропускную способность этой сети с большим количеством соединений других ее пользователей. Следствием этого факта являются достоинства и недостатки сервиса VPN. Достоинством для провайдера является то, что с помощью сети с коммутацией пакетов он может обслужить большее число клиентов, это вытекает из самой природы сети с ее статическим мультиплексированием трафика клиентов (как вы знаете из материала главы 2, в сети с коммутацией каналов пропускная способность канала всегда расходуется не полностью, особенно если трафик, передаваемый по каналу, имеет значительные пульсации, а в нашем случае мы рассматриваем соединение компьютерных сетей клиентов, для которых характерен именно такой трафик). Для потребителей данной услуги преимуществом является более низкая ее стоимость, чем в случае услуги выделенных каналов, так как себестоимость услуг сетей с коммутацией пакетов обычно существенно ниже, чем сетей с коммутацией каналов при равной скорости соединений. Другим преимуществом является доступность услуги: многие провайдеры услуг Интернета предоставляют также и услуги VPN, так что организация, получающая доступ в Интернет с помощью такого провайдера, может дополнительно воспользоваться услугой VPN, которая конфигурируется как дополнительное логическое соединение. Кроме того, *у самого клиента существует возможность организовать виртуальную частную сеть своими силами, для этого достаточно иметь обычный доступ в Интернет.*

Сервис виртуальных частных сетей может быть реализован различными способами и с различной степенью приближения к сервису частных сетей на выделенных каналах, который он эмулирует. Ввиду важности этого сервиса мы рассмотрим его в отдельном разделе (см. далее раздел «Виртуальные частные сети»).

Доступ в Интернет

С появлением Интернета ситуация в мире принципиально изменилась, так как появилась глобальная публичная сеть с коммутацией пакетов, аналог всемирной телефонной сети. Как и в случае телефонной сети, любому индивидуальному пользователю или организации можно подключиться к такой сети и получить возможность оперативно связываться с любым другим ее абонентом. Это обстоятельство является принципиальным отличием от услуг виртуальных частных сетей, которые соединяют своих пользователей выборочно. Так как Интернет представляет собой объединение всех сетей отдельных операторов связи без ограничения взаимодействия между этими сетями (есть редкие исключения в некоторых странах), то услуга доступа в Интернет реализуется как услуга доступа пользователя (его сети или отдельного компьютера) к сети некоторого оператора связи. Операторы связи выступают в данном случае в роли поставщиков услуг Интернета. В результате пользователь получает доступ к любому компьютеру, который аналогичным образом получил доступ к сети другого оператора. Протоколы IP обеспечивают полную связность IP-сетей операторов связи между собой, никаких дополнительных усилий по доступу отдельных

пользователей к узлам Интернета от оператора связи не требуется; такая связь каждого с каждым является принципом организации Интернета.

Рисунок 19.3 иллюстрирует возможности, которые получает потребитель услуги доступа в Интернет. Здесь сети операторов 1, 2 и 3 являются частью Интернета, то есть операторы этих сетей являются по совместительству поставщиками услуг Интернета. Это значит, что они имеют соглашения о передаче трафика Интернета между собой и некоторыми другими провайдерами, сети которых на рисунке не показаны. Сети этих провайдеров физически связаны, а пограничные маршрутизаторы сетей получают от своих соседей по протоколу BGP всю необходимую информацию о сетях, входящих в Интернет, поэтому могут правильно маршрутизировать любой запрос на взаимодействие с любым узлом Интернета. За счет этого клиент 1 может обратиться к любому из серверов, подключенных к Интернету, а также взаимодействовать с другими клиентами Интернета по одноранговым протоколам, например по протоколу IP-телефонии. Сама услуга доступа в Интернет является транспортной, то есть она сама по себе не предоставляет никаких прикладных сервисов, таких как веб-сервис или сервис IP-телефонии. Эти прикладные сервисы (рассматриваемые в главе 23) работают поверх службы доступа в Интернет, и для самого транспорта Интернета они прозрачны (говорят, что транспорт Интернета нейтрален к прикладным услугам, эта нейтральность является одним из принципов организации Интернета).

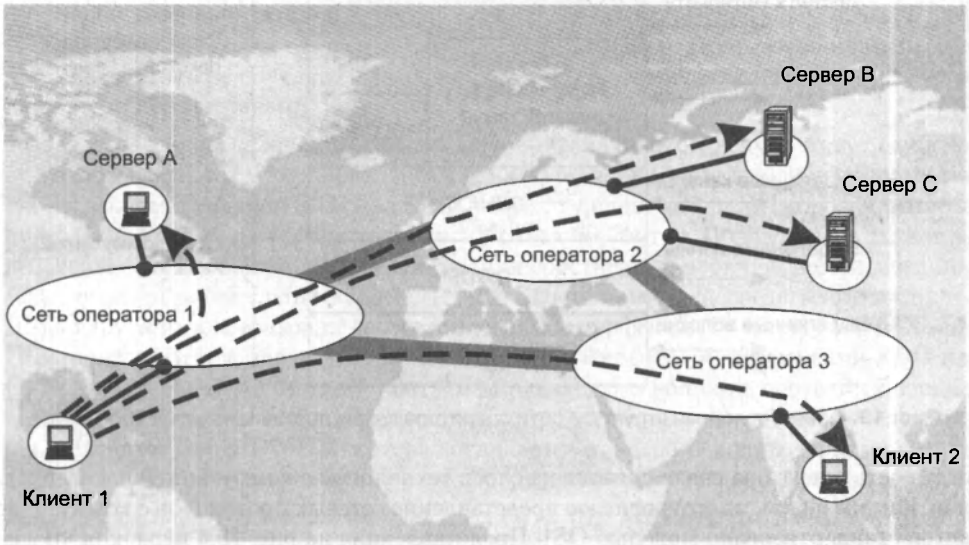


Рис. 19.3. Услуга доступа в Интернет

Интернет может использоваться и для предоставления услуг виртуальных частных сетей. В этом случае необходимо каким-то образом *подавить встроенную в Интернет возможность каждого общаться с каждым*. Чаще всего такое ограничение реализуется конечными пользователями Интернета — организациями или индивидуальными пользователями, а не поставщиками услуг Интернета. Хотя последний вариант также возможен, он требует от провайдера значительных усилий по защите пользователей виртуальной частной сети от остальной части пользователей Интернета.

Традиционная телефония

Для многих операторов связи (особенно крупных национальных компаний, таких как AT&T или BT) предоставление услуг традиционной телефонии по-прежнему остается очень важной частью их бизнеса. Этот бизнес требует наличия у оператора глобальной сети телефонных коммутаторов, объединенных физическими каналами связи.

Многослойная сеть оператора связи

Для предоставления услуг всех перечисленных типов оператор связи должен иметь многослойную сеть. Каждый слой такой сети может выполнять две функции:

- предоставление услуг конечным пользователям;
- поддержка функций вышележащих уровней сети оператора.

Обобщенная структура слоев типичной сети оператора связи, который также играет роль поставщика услуг Интернета, показана на рис. 19.4.

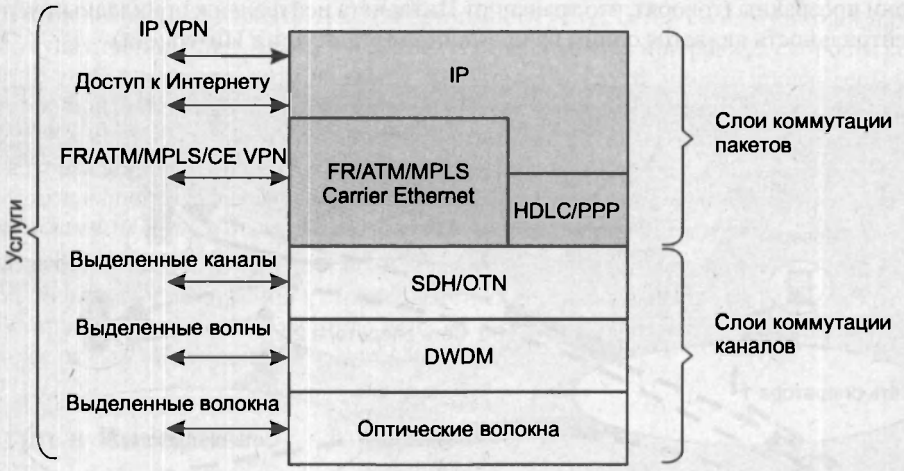


Рис. 19.4. Многослойная структура сети оператора связи/поставщика услуг Интернета

Каждая сеть оператора связи состоит из слоев технологий с коммутацией пакетов и каналов. Как мы знаем, многоуровневое представление сетевых протоколов с коммутацией пакетов стандартизовано моделью OSI. Представленная на рис. 19.4 иерархия уровней соответствует этой модели, если принять во внимание два обстоятельства:

- мы рассматриваем транспортные технологии глобальных сетей, поэтому наш интерес заканчивается слоем протокола IP, то есть сетевым уровнем, который является высшим обязательным уровнем протоколов транспортной подсистемы (мы рассматривали этот вопрос в разделе «Распределение протоколов по элементам сети» главы 4);
- физический уровень модели OSI в сетях операторов связи представлен несколькими слоями, соответствующими технологиям первичных сетей.

Услуги и технологии физического уровня

Особенностью глобальных сетей является структура физического уровня: он гораздо сложнее, чем физический уровень локальных сетей, где на этом уровне используются только кабели. В глобальных сетях для создания канала между двумя коммутаторами или маршрутизаторами, как правило, применяются устройства первичных сетей, такие как мультиплексоры или кросс-коннекторы сетей PDH, SDH, OTN или DWDM (подробно технологии первичных сетей рассматриваются в главе 11).

Первоначально технологии первичных сетей предназначались только для внутренних целей операторов связи в качестве гибкого средства соединения телефонных коммутаторов, то есть для гибкого создания каналов между их собственными коммутаторами, изначально телефонными, а потом и пакетными. Постепенно с ростом популярности компьютерных сетей технологии первичных сетей стали применяться для предоставления транспортных услуг конечным пользователям.

Именно поэтому на рис. 19.4 показаны три типа услуг, которые предоставляются операторами связи с помощью трех нижних слоев их сети:

- *Услуга выделенных оптических волокон.* Эта услуга чаще всего оказывается одним оператором, обладающим развитой кабельной инфраструктурой со свободными оптическими кабелями, или волокнами, другому оператору, который затем строит на этих волокнах собственную первичную сеть, соединяя с помощью волокон мультиплексоры DWDM/OTN или SDH. Волокна, сдаваемые в аренду, часто называют **темными волокнами** (dark fibre), так как они не подключены к оборудованию передачи данных и не «подсвечены» лазерными передатчиками.
- *Услуга выделенных волновых каналов.* Потребителями этой услуги могут быть как операторы связи, так и корпоративные пользователи. Обычно такая услуга предоставляется в формате кадров OTN или SDH высшего уровня иерархии скорости, который в настоящее время для обеих технологий равен 40 Гбит/с. Пользователь может задействовать волновой канал для построения собственной первичной сети, соединяя таким образом свои мультплексоры OTN или SDH, а может непосредственно соединить IP-маршрутизаторы, имеющие соответствующие интерфейсы (OTN или SDH). Обычно IP-маршрутизаторы обладают так называемыми «серыми» интерфейсами SDH или OTN; это означает, что они работают с неокрашенными волнами, соответствующими центру окна прозрачности, например с волной 1310 нм. Для того чтобы использовать определенную волну DWDM, которая отличается от «серой» волны, например волну 1528,77 нм, необходим **транспондер** — устройство преобразования длин волн.
- *Услуга выделенного соединения по протоколу OTN, SDH или PDH.* Это наиболее традиционная услуга оператора связи, когда пользователь берет в аренду выделенные каналы нужной ему скорости, например каналы со скоростями 34 (E3) и 622 Мбит/с (STM-4). Эти каналы соединяют географически разнесенные локальные сети предприятия, и на них пользователь строит свою корпоративную компьютерную сеть (напомним, что она называется в таком случае частной), соединяя этими каналами свои IP-маршртизаторы или FR-коммутаторы. В последнее время стала популярной такая услуга, как *выделенный канал на 1 Гбит/с с интерфейсом Ethernet.* Как вы знаете, скорость 1 Гбит/с не является стандартной для технологий первичных сетей, однако монополия Ethernet в локальных сетях привела к ситуации, когда выделенные каналы все чаще служат для соединения пограничных устройств клиентов с интерфейсами Ethernet. Поэтому появление такой услуги, как канал со скоростью 1 Гбит/с, явилось ответом на потреб-

ности пользователей, при этом пограничный мультиплексор SDH или OTN оснащается интерфейсом Ethernet, а принимаемые кадры Ethernet затем упаковываются мультиплексором в кадры SDH или OTN и отправляются по соединению сети SDH или OTN, арендованному пользователем (также могут применяться кадры GFP, получаемые универсальным методом кадрирования, а в сетях SDH еще и методы мультплексирования VCAT, позволяющие более эффективно расходовать емкость контейнеров).

Нужно понимать, что рис. 19.4 иллюстрирует общий случай структуры физического уровня сети оператора связи. В конкретных случаях отдельные элементы этой общей структуры могут отсутствовать. Например, как уже отмечалось, оператор может не иметь собственной инфраструктуры оптических кабелей, так как прокладывать кабели под землей или на опорах — дело весьма дорогостоящее и трудоемкое. Технология SDH начала постепенно вытесняться технологией OTN, так что ее поддержание у операторов связи в ближайшем будущем не очевидно (кроме того, сегодня в качестве пакетной замены SDH рассматривается Ethernet операторского класса).

Услуги и технологии пакетных уровней

Транспортная система сетей операторов связи включает два уровня технологий, которые относятся к канальному и сетевому уровням модели OSI.

На сетевом уровне сегодня применяется лишь протокол IP, все остальные (такие как IPX или DECnet) благодаря успехам Интернета сошли со сцены. IP является обязательным протоколом, так как он нужен оператору связи/поставщику услуг Интернета как для предоставления доступа в Интернет своим клиентам, так и для взаимодействия с сетями других операторов связи/поставщиков услуг.

Более сложная ситуация наблюдается на канальном уровне. Как видно из рис. 19.4, здесь могут использоваться разные технологии (на рисунке они объединены в два прямоугольника разной высоты, что символизирует свойства двух групп технологий канального уровня). Первая группа технологий, в которую входят технологии ATM, Frame Relay, MPLS и Carrier Ethernet, отличается тем, что с их помощью можно построить сеть, выполняющую коммутацию пакетов (кадров, ячеек — термины могут быть разными, но суть в том, что эти технологии подразумевают наличие коммутаторов, способных продвигать данные на основе адресной информации той или иной технологии).

Главной особенностью технологий второй группы, в которую входят протоколы HDLC и PPP, является то, что эти *технологии предназначены для работы на двухточечных соединениях*. Это означает, что они могут передавать данные только между двумя непосредственно соединенными интерфейсами, но не далее. В этих технологиях не используются уникальные адреса конечных узлов, так как их задача очень проста — передача кадра непосредственному соседу. Можно сказать, что это технологии интерфейсов, так как они действительно реализуются в интерфейсах маршрутизаторов или конечных узлов — компьютеров. При этом задачу коммутации пакетов решает маршрутизатор на основе IP-адресов, а интерфейсная технология требуется только для доставки IP-пакета соседнему маршрутизатору. Меньшая высота прямоугольника отражает более бедную функциональность этой группы протоколов.

Протоколы первой группы могут служить как для внутренних целей, обеспечивая IP-маршрутизаторы своими соединениями, так и для предоставления услуг пользователям.

Оба этих варианта использования технологий канального уровня с коммутацией каналов иллюстрирует рис. 19.5.

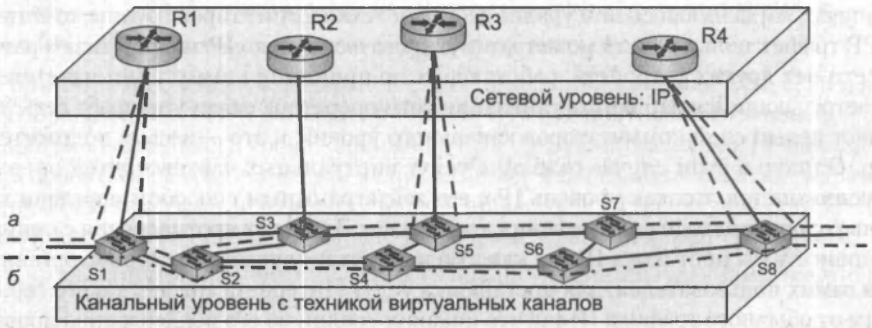


Рис. 19.5. Использование канального уровня для организации соединений между маршрутизаторами

В этом примере в сети имеется 4 маршрутизатора и 8 коммутаторов канального уровня, которые поддерживают одну из технологий виртуальных каналов (в данном случае не принципиально, какую именно). Маршрутизаторы связаны между собой через слой коммутаторов, непосредственных физических связей между маршрутизаторами нет. Для связи маршрутизаторов используется четыре виртуальных канала, как показано на рис. 19.6.

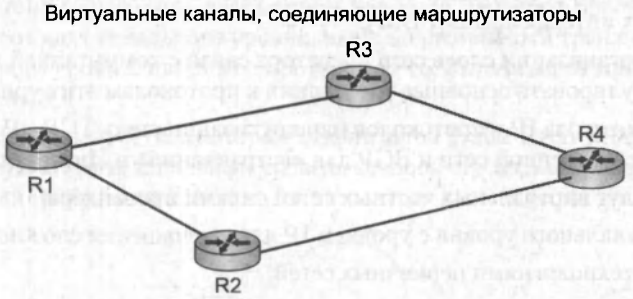


Рис. 19.6. Соединение маршрутизаторов через четыре виртуальных канала

При обслуживании трафика доступа в Интернет он проходит через маршрутизаторы в соответствии с имеющимися между ними связями и таблицами маршрутизации. На рис. 19.5 путь такого трафика показан пунктирной линией, помеченной буквой *а*. Реализация связей между маршрутизаторами с помощью виртуальных каналов обеспечивает:

- высокий уровень управляемости потоков данных, то есть позволяет контролировать загрузку каналов и поддерживать хорошее качество обслуживания пользовательского трафика;
- мониторинг соединений, а это важно для провайдера платных услуг, работающего на основе контрактов с пользователями.

Однако в том случае, когда провайдеру нужно объединить две сети пользователя с помощью услуги виртуальной частной сети, это проще сделать с помощью слоя канального уровня без помощи сетевого уровня. На рис. 19.5 прохождение трафика услуги виртуальной частной сети через сеть провайдера показано штрих-пунктирной линией, помеченной буквой *б*.

В том случае, когда на канальном уровне работают технологии второй группы, то есть HDLC или PPP, трафик пользователя может коммутироваться только IP-маршрутизаторами[1], так как в сети нет других устройств, работающих по принципу коммутации пакетов. Такой также встречающийся вариант организации сети оператора связи упрощает сеть, так как устраняет целый слой коммутаторов канального уровня, и это — весьма положительный фактор. Однако в этом случае оказание услуг виртуальных частных сетей оператором связи усложняется, так как уровень IP с его дейтаграммным способом передачи данных не очень хорошо подходит для решения этой задачи. Здесь нет противоречия с популярностью сервиса VPN протокола IP, так как в большинстве случаев этот сервис организуется силами самих пользователей; для поставщика услуг Интернета трафик такого сервиса не отличим от обычного трафика IP, так что никаких усилий по его поддержанию провайдеру прикладывать не нужно. Однако столь высоких характеристик в плане гарантии пропускной способности соединений VPN, которые могут быть достигнуты в случае реализации сервиса провайдером на канальном уровне, пользовательский сервис VPN достигнуть не может.

Пакетные слои могут взаимодействовать с различными слоями первичной сети для получения физических соединений между маршрутизаторами или коммутаторами. Совсем не обязательно взаимодействовать с самым верхним слоем первичной сети, например со слоем PDH или SDH. В том случае, когда маршрутизаторам или коммутаторам необходимы высокоскоростные соединения, можно их организовывать с помощью нижних слоев первичной сети, например, с помощью слоя DWDM (мы уже упоминали о маршрутизаторах, поддерживающих интерфейсы DWDM).

Анализ услуг и организации слоев сети оператора связи с коммутацией пакетов дает возможность сформулировать основные требования к протоколам этих уровней:

- поддержка протокола IP и протоколов маршрутизации стека TCP/IP (OSPF, IS-IS для организации собственной сети и BGP для «встраивания» в Интернет);
- поддержка услуг виртуальных частных сетей силами провайдера;
- интеграция канального уровня с уровнем IP для уменьшения сложности сети;
- интеграция с технологиями первичных сетей.

Туннелирование

Сети операторов связи могут также предоставлять услуги виртуальных частных сетей на основе техники туннелирования. Эта техника уже рассматривалась нами на частном примере туннелирования трафика IPv6 через IPv4-сеть. Так как техника туннелирования весьма распространена, здесь мы рассмотрим ее с общих позиций.

Туннелирование, или *инкапсуляция*, — это нестандартный (отличающийся от принятого в модели OSI порядка) способ инкапсуляции пакетов некоторого протокола двух объединяемых сетей или узлов в пакеты протокола транзитной сети на ее границе и передача пакетов объединяемых сетей через транзитную сеть. Туннелирование применяется в тех

---

[1] Мы здесь использовали термин «коммутация» как обобщенный термин, то есть в том же смысле, в котором он употреблялся в разделе «Обобщенная задача коммутации» главы 2. В этом контексте более привычный для описания работы сетевого уровня термин «маршрутизация» является частным случаем коммутации пакетов.

случаях, когда транзитная сеть либо не поддерживает протокол объединяемых сетей, либо стремится изолировать транзитную сеть от объединяемых сетей.

Данное описание подходит к стандартной схеме, описанной в модели OSI, если под протоколом объединяемых сетей понимать протокол IP, а под протоколом транзитной сети — любой протокол канального уровня, например Ethernet. Действительно, IP-пакеты могут инкапсулироваться на границе сети в кадры Ethernet и передаваться в этих кадрах через транзитную сеть Ethernet в неизменном виде. А при выходе из транзитной сети IP-пакеты извлекаются из кадров Ethernet и дальше уже обрабатываются маршрутизатором.

Для того чтобы понять, в чем нестандартность инкапсуляции, сначала заметим, что в этом процессе принимают участие три типа протоколов:

- протокол-пассажир;
- несущий протокол;
- протокол инкапсуляции.

При стандартной работе составной сети, описанной в модели OSI (и повсеместно применяемой на практике), протоколом-«пассажиром» является протокол IP, а несущим протоколом — один из протоколов канального уровня отдельных сетей, входящих в составную сеть, например Frame Relay или Ethernet. Протоколом инкапсуляции также является протокол IP, для которого функции инкапсуляции описаны в стандартах RFC для каждой существующей технологии канального уровня.

При туннелировании протоколом-пассажиром является протокол объединяемых сетей, это может быть протокол канального уровня, не поддерживаемый транзитной сетью, или же протокол сетевого уровня, например протокол IPv6, отличный от протокола сетевого уровня транзитной сети.

На рис. 19.7 показан пример сети, в которой трафик сетей Frame Relay передается по туннелю через транзитную IP-сеть, канальный уровень которой эту технологию не поддерживает, так как построен на технологии Ethernet.

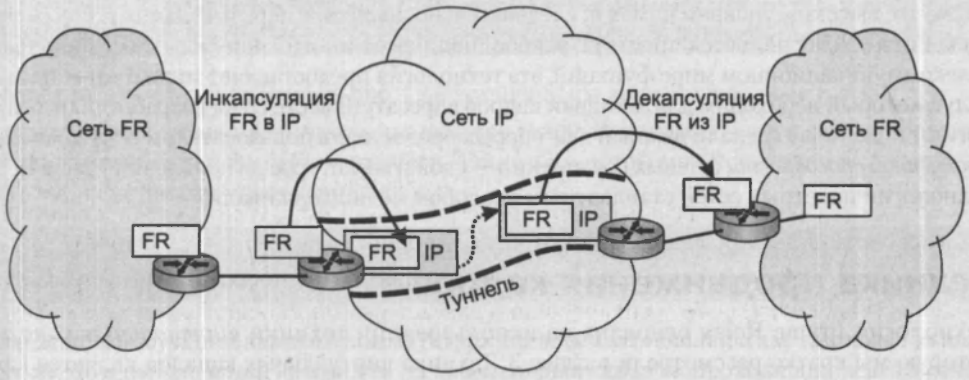


Рис. 19.7. Туннелирование трафика Frame Relay через IP-сеть

Таким образом, протоколом-пассажиром является протокол FR, а несущим протоколом — протокол IP. Пакеты протокола-пассажира помещаются в поле данных пакетов несущего протокола с помощью протокола инкапсуляции. Инкапсуляция FR-кадров в IP-пакеты не является стандартной операцией для IP-маршрутизаторов. Это дополнительная для

маршрутизаторов функция описывается отдельным стандартом и должна поддерживаться пограничными маршрутизаторами транзитной сети, если мы хотим организовать такой туннель.

Инкапсуляцию выполняет пограничное устройство (обычно маршрутизатор или шлюз), которое располагается на границе между исходной и транзитной сетями. Пакеты протокола-пассажира при транспортировке их по транзитной сети никак не обрабатываются. Извлечение пакетов-пассажиров из несущих пакетов выполняет второе пограничное устройство, которое находится на границе между транзитной сетью и сетью назначения. Пограничные маршрутизаторы указывают в IP-пакетах, переносящих трафик туннеля, свои IP-адреса в качестве адресов назначения и источника.

В связи с популярностью Интернета и стека TCP/IP ситуация, когда несущим протоколом транзитной сети обычно выступает протокол IP, а протоколом-пассажиром — некоторый канальный протокол, является очень распространенной. Вместе с тем применяются и другие схемы инкапсуляции, такие как инкапсуляция IP в IP, Ethernet в MPLS, Ethernet в Ethernet. Подобные схемы инкапсуляции нужны не только для того, чтобы согласовать транспортные протоколы, но и для других целей, например для шифрования исходного трафика или для изоляции адресного пространства транзитной сети провайдера от адресного пространства пользовательских сетей.

Технология Frame Relay

История стандарта

Пакетная технология глобальных сетей **Frame Relay** появилась в конце 80-х годов в связи с распространением высокоскоростных и надежных цифровых каналов технологий PDH и SDH. До этого основной технологией глобальных сетей являлась технология X.25, сложный стек которой был рассчитан на низкоскоростные аналоговые каналы, отличавшиеся к тому же высоким уровнем помех и, следовательно, ошибок в передаче данных. Особенностью Frame Relay является простота; освободившись от многих ненужных в современном телекоммуникационном мире функций, эта технология предоставляет только тот минимум услуг, который необходим для доставки кадров адресату. Вместе с тем разработчики технологии Frame Relay сделали важный шаг вперед, предоставив пользователям сети *гарантию пропускной способности* сетевых соединений — свойство, которое до появления Frame Relay технологии пакетных сетей стандартным способом не поддерживали.

Техника продвижения кадров

Технология Frame Relay основана на использовании техники *виртуальных каналов*, которую мы кратко рассмотрели в главе 3. Техника виртуальных каналов является компромиссом между неопределенностью дейтаграммного способа продвижения пакетов, используемого, например, в сетях Ethernet и IP, и жесткостью коммутации каналов, которая свойственна технологиям первичных и телефонных сетей.

Рассмотрим технику виртуальных каналов сетей Frame Relay на примере сети, изображенной на рис. 19.8.

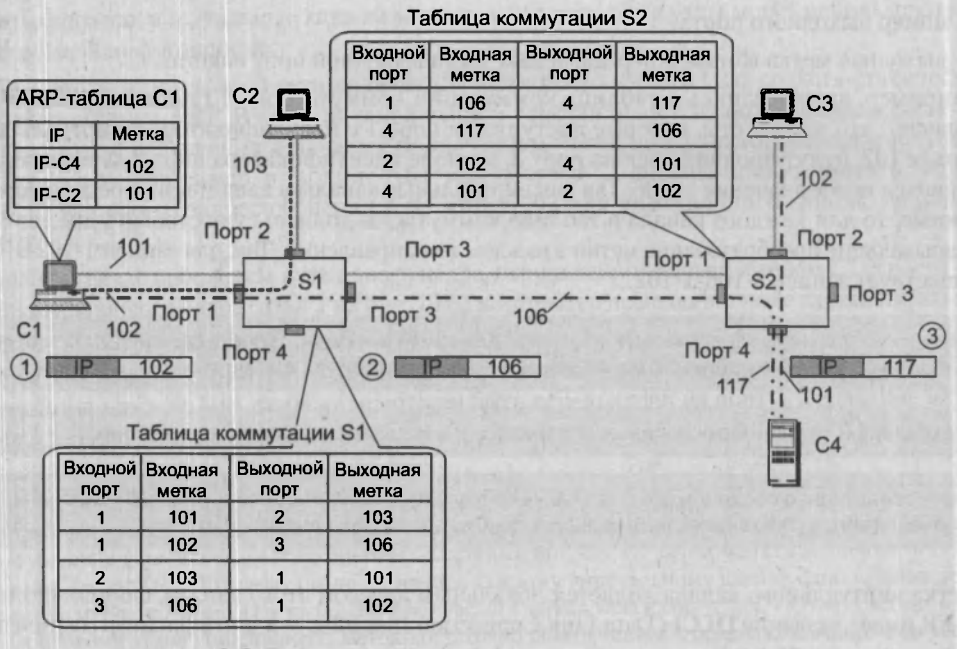


Рис. 19.8. Продвижение кадров вдоль виртуальных каналов FR

Для того чтобы конечные узлы сети — компьютеры *C*1, *C*2, *C*3 и сервер *C*4 — могли обмениваться данными, в сети необходимо предварительно проложить виртуальные каналы. В нашем примере установлено три таких канала — между компьютерами *C*1 и *C*2 через коммутатор *S*1; между компьютером *C*1 и сервером *C*4 через коммутаторы *S*1 и *S*2; между компьютером *C*3 и сервером *C*4 через коммутатор *S*2.

Виртуальные каналы Frame Relay могут быть как **однонаправленными** (то есть способными передавать кадры только в одном направлении), так и **двунаправленными**.

Будем считать, что в примере на рис. 19.8 установлены двунаправленные каналы.

Процедура установления виртуальных каналов Frame Relay заключается в формировании таблиц коммутации в коммутаторах сети. Такие процедуры могут выполняться как вручную, так и системами управления сетью.

Виртуальные каналы Frame Relay относятся к типу **постоянных виртуальных каналов** (Permanent Virtual Circuit, PVC), они заранее устанавливаются по командам оператора сети.

В таблице коммутации каждого коммутатора должны быть сделаны две записи (для каждого из двух направлений) о каждом из виртуальных каналов, проходящих через данный коммутатор.

Запись таблицы коммутации состоит из четырех основных полей, каковыми являются:

- номер входного порта канала;
- входная метка канала в поступающих на входной порт пакетах;

- номер выходного порта;
- выходная метка канала в передаваемых через выходной порт пакетах.

Например, вторая запись в таблице коммутации коммутатора *S*1 (запись 1-102-3-106) означает, что все пакеты, которые поступят на порт 1 с идентификатором виртуального канала 102, будут продвигаться на порт 3, а в поле идентификатора виртуального канала появится новое значение — 106. Так как виртуальные каналы в нашем примере двунаправленные, то для каждого канала в таблице коммутации должно существовать две записи, описывающие преобразование метки в каждом из направлений. Так, для записи 1-102-3-106 существует запись 3-106-1-102.

Метки виртуального канала имеют *локальное* для коммутатора и его порта значение, то есть они никаким образом не принимаются во внимание на портах других коммутаторов.

Комбинации «метка-порт» должны быть *уникальными* в пределах одного коммутатора.

Непосредственно соединенные порты двух коммутаторов должны использовать *согласованные* значения меток для каждого виртуального канала, проходящего через эти порты.

Метка виртуального канала является локальным адресом этого канала, формально метка FR имеет название **DLCI** (Data Link Connection Identifier — идентификатор соединения уровня канала данных).

Метки DLCI переносятся кадрами FR; формат такого кадра показан на рис. 19.9.

Рис. 19.9. Формат кадра FR

Поле **DLCI** состоит из 10 бит, что позволяет задействовать до 1024 виртуальных соединений. Поле DLCI может занимать и большее число разрядов — этим управляют признаки расширения адреса EA0 и EA1 (аббревиатура EA как раз и означает Extended Address, то есть расширенный адрес). Если бит расширения адреса установлен в ноль, то признак называется EA0 и означает, что в следующем байте имеется продолжение поля адреса, а если бит расширения адреса равен 1, то поле называется EA1 и означает окончание поля адреса. Десятиразрядный формат DLCI является основным, но при использовании трех байтов для адресации поле DLCI имеет длину 16 бит, а при использовании четырех байтов — 23 бита.

Поле данных может иметь размер до 4056 байт.

Поле **C/R** переносит признак команды (Command) или ответа (Response). Этот признак является унаследованным от протоколов X.25 и в операциях FR не используется.

Поля **DE** (Discard Eligibility), **FECN** (Forward-explicit congestion notification) и **BECN** (Backward-explicit congestion notification) используются протоколом FR для оповещения коммутаторов сети FR о возможности отбрасывания кадров (DE), а также о перегрузке в сети (FECN и BECN).

После того как виртуальные каналы установлены, конечные узлы могут использовать их для обмена информацией.

Для этого администратор сети должен для каждого конечного узла создать статические записи таблицы ARP. В каждой такой записи устанавливается соответствие между IP-адресом узла назначения и начальным значением метки виртуального канала, ведущего к этому узлу. Например, в таблице ARP компьютера *C*1 должна присутствовать запись, отображающая IP-адрес сервера *C*4 на метку 102 для виртуального канала, ведущего к серверу *C*4.

Давайте сейчас проследим путь одного кадра, отправленного компьютером *C*1 серверу *C*4. При отправлении кадра (этап 1 на рис. 19.8) компьютер помещает в поле адреса начальное значение метки 102, взятое из его таблицы ARP.

Коммутатор *S*1, получив на порт 1 кадр с меткой 102, просматривает свою таблицу коммутации и находит, что такой кадр должен быть переправлен на порт 3, а значение метки в нем должно быть заменено на 106.

ПРИМЕЧАНИЕ

Операция по замене метки (label swapping) характерна для всех технологий, использующих технику виртуальных каналов. Может возникнуть законный вопрос: «А зачем менять значение метки на каждом коммутаторе? Почему бы не назначить каждому виртуальному каналу одно неизменяемое значение метки, которая бы играла роль физического адреса узла назначения?» Ответ состоит в том, что в первом случае уникальность меток достаточно обеспечивать в пределах каждого отдельного порта, а во втором — в пределах всей сети, что гораздо сложнее, так как требует наличия в сети централизованной службы назначения меток.

В результате действий коммутатора *S*1 кадр отправляется через порт 3 к коммутатору *S*2 (этап 2). Коммутатор *S*2, используя свою таблицу коммутации, находит соответствующую запись, заменяет значение метки на 117 и отправляет кадр узлу назначения — серверу *C*4. На этом обмен заканчивается, а при отправке ответа сервер *C*4 задействует метку 117 как адрес виртуального канала, ведущего к компьютеру *C*1.

Как видно из этого описания, коммутация выполняется очень экономично, так как преобразования передаваемых кадров минимальны — они сводятся только к замене значения метки. В кадрах указывается только адрес назначения, роль которого в сетях Frame Relay играет метка. В качестве адреса отправителя может быть использовано последнее значение метки, оно однозначно определяет путь в обратном направлении по виртуальному каналу, соединяющему получателя и отправителя.

Гарантии пропускной способности

Сети Frame Relay создавались для оказания коммерческих услуг операторов связи по передаче компьютерного трафика. Одной из новых и очень привлекательных для клиентов услуг Frame Relay стала поддержка гарантий пропускной способности виртуальных соединений. Для каждого виртуального соединения в технологии Frame Relay определяется несколько параметров, связанных со скоростью передачи данных.

- **Согласованная скорость передачи данных** (Committed Information Rate, CIR) — гарантированная пропускная способность соединения; фактически сеть гарантирует передачу данных пользователя со скоростью предложенной нагрузки, если эта скорость не превосходит CIR.

- **Согласованная величина пульсации** (Committed Burst Size, Bc) – максимальное количество байтов, которое сеть будет передавать от данного пользователя за интервал времени *T*, называемый временем пульсации, соблюдая согласованную скорость CIR.
- **Дополнительная величина пульсации** (Excess Burst Size, Be) – максимальное количество байтов, которое сеть будет пытаться передать сверх установленного значения Bc за интервал времени *T*.

Второй параметр пульсации Be позволяет оператору сети дифференцированно обрабатывать кадры, которые не укладываются в профиль CIR. Обычно кадры, которые приводят к превышению пульсации Bc, но не превышают пульсации Bc + Be, сетью не отбрасываются, а обслуживаются, но без гарантий по скорости CIR. Для запоминания факта нарушения в кадрах Frame Realy используется поле DE. И только если превышен порог Bc + Be, кадры отбрасываются.

Если приведенные величины определены, то время *T* определяется следующей формулой:

$$T = Bc/CIR.$$

Можно рассматривать значения CIR и *T* в качестве варьируемых параметров, тогда производной величиной станет пульсация Bc. Обычно для контроля пульсаций трафика выбирается время *T*, равное 1–2 секундам при передаче компьютерных данных и в диапазоне десятков-сотен миллисекунд при передаче голоса.

Соотношение между параметрами CIR, Bc, Be и *T* иллюстрирует рис. 19.10 (*R* – скорость в канале доступа; f_1–f_5 – кадры).

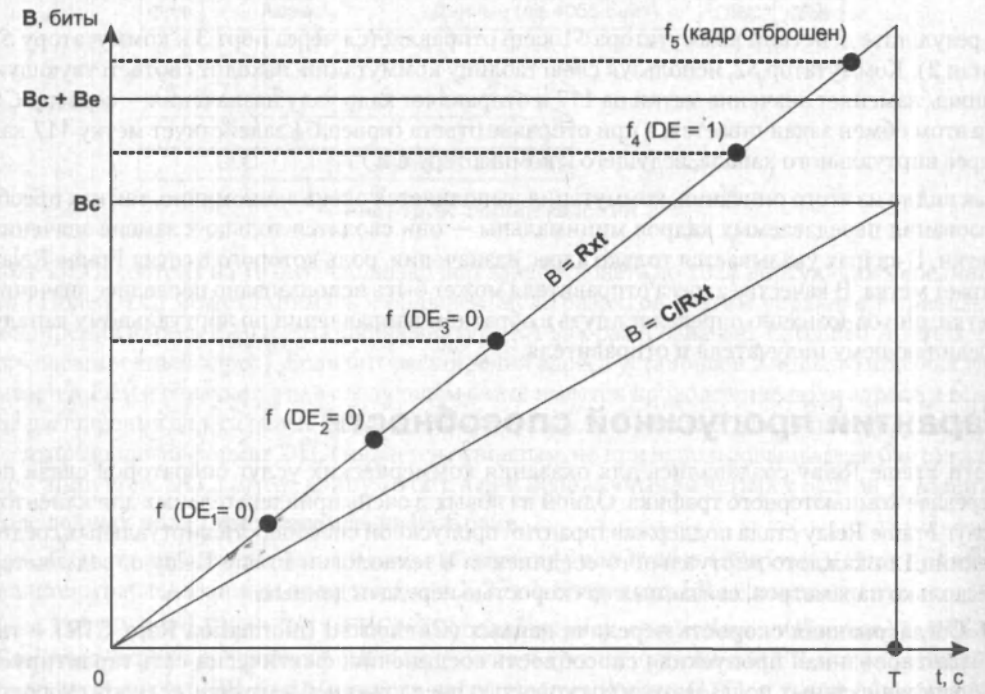


Рис. 19.10. Реакция сети на поведение пользователя

Работа сети описывается двумя линейными функциями, показывающими зависимость количества переданных битов от времени: $B = R \times t$ и $B = \text{CIR} \times t$. Средняя скорость поступления данных в сеть составила на этом интервале R бит/с, и она оказалась выше CIR. На рисунке представлен случай, когда за интервал времени T в сеть по виртуальному каналу поступило 5 кадров. Кадры f_1, f_2 и f_3 доставили в сеть данные, суммарный объем которых не превысил порог Bc, поэтому эти кадры ушли дальше транзитом с признаком DE = 0. Данные кадра f_4, прибавленные к данным кадров f_1, f_2 и f_3, уже превысили порог Bc, но еще не достигли порога Bc + Be, поэтому кадр f_4 также ушел дальше, но уже с признаком DE = 1. Данные кадра f_5, прибавленные к данным предыдущих кадров, превысили порог Bc + Be, поэтому этот кадр был удален из сети.

На рис. 19.11 приведен пример сети Frame Relay с пятью удаленными региональными отделениями корпорации. Обычно доступ к сети осуществляется по каналам с пропускной способностью, большей чем CIR. Однако при этом пользователь платит не за пропускную способность канала, а за заказанные величины CIR, Bc и Be. Так, при применении в качестве линии доступа канала T1 и заказа обслуживания со скоростью CIR, равной 128 Кбит/с, пользователь будет платить только за скорость 128 Кбит/с, а скорость канала T1 в 1,5 Мбит/с окажет влияние на верхнюю границу возможной пульсации Bc + Be.

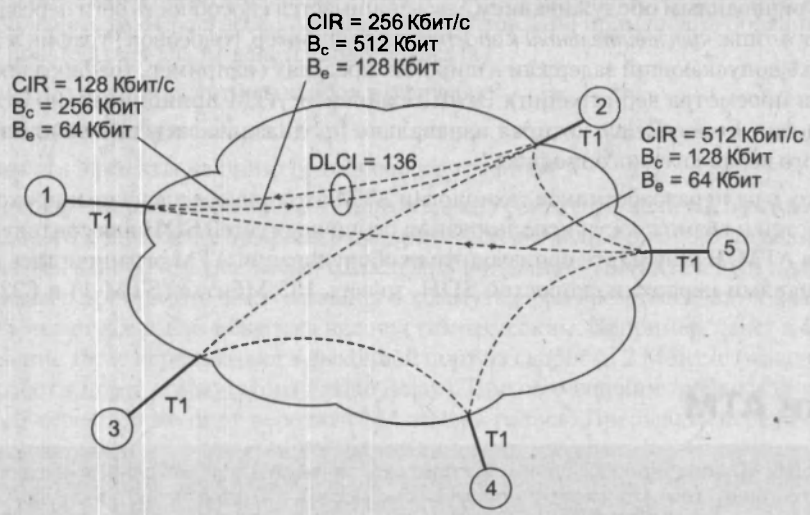


Рис. 19.11. Пример обслуживания в сети Frame Relay

Параметры качества обслуживания могут быть разными для разных направлений виртуального канала. Так, на рисунке абонент 1 соединен с абонентом 2 виртуальным каналом с меткой 136. При направлении от абонента 1 к абоненту 2 канал имеет среднюю скорость 128 Кбит/с с пульсациями Bc = 256 Кбит (интервал T составил 1 с) и Be = 64 Кбит. А при передаче кадров в обратном направлении средняя скорость уже может достигать значения 256 Кбит/с с пульсациями Bc = 512 Кбит и Be = 128 Кбит.

Технология Frame Relay получила большое распространение в сетях операторов связи в 90-е годы благодаря простоте и возможности гарантировать клиентам пропускную способность соединений. Тем не менее в последнее время популярность услуг Frame Relay резко упала, в основном это произошло из-за появления технологии MPLS, которая, так же как и Frame Relay, основана на технике виртуальных каналов и может гарантировать

пропускную способность пользовательских соединений. Решающим преимуществом MPLS является ее тесная интеграция с технологией IP, за счет этого провайдерам легче формировать новые комбинированные услуги. Кроме того, функциональность MPLS поддерживается сегодня практически всеми маршрутизаторами среднего и высшего класса, так что применение MPLS не требует установки в сети отдельных коммутаторов.

Более подробную информацию вы можете найти на сайте www.olifer.co.uk в разделе «Технология Frame Relay».

Технология ATM

Асинхронный режим передачи (Asynchronous Transfer Mode, ATM) — это технология, основанная на установлении *виртуальных каналов* и предназначенная для использования в качестве единого универсального транспорта нового поколения сетей с интегрированным обслуживанием.

Под **интегрированным обслуживанием** здесь понимается способность сети передавать трафик разного типа: *чувствительный к задержкам* (например, голосовой) трафик и *эластичный*, то есть допускающий задержки в широких пределах (например, трафик электронной почты или просмотра веб-страниц). Этим технология ATM принципиально отличается от технологии Frame Relay, которая изначально предназначалась только для передачи эластичного компьютерного трафика.

Кроме того, в цели разработчиков технологии ATM входило обеспечение широкой иерархии скоростей и возможности использования первичных сетей SDH для соединения коммутаторов ATM. В результате производители оборудования ATM ограничились первыми двумя уровнями иерархии скоростей SDH, то есть 155 Мбит/с (STM-1) и 622 Мбит/с (STM-4).

Ячейки ATM

В технологии ATM для переноса данных используются **ячейки**. Принципиально ячейка отличается от кадра только тем, что имеет, во-первых, *фиксированный*, во-вторых, *небольшой* размер. Длина ячейки составляет 53 байта, а поля данных — 48 байт. Именно такие размеры позволяет сети ATM передавать чувствительный к задержкам аудио- и видеотрафик с необходимым уровнем качества.

Главным свойством ATM, которое отличает ее от других технологий, является комплексная поддержка параметров QoS для всех основных видов трафика.

Для достижения этого свойства разработчики ATM тщательно проанализировали все типы трафика и провели его классификацию. Мы уже познакомились с этой классификацией в главе 7, когда рассматривали требования различных приложений к QoS. Напомним, что в ATM весь трафик разбивается на 5 классов, A, B, C, D и X. Первые четыре класса представляют трафик типовых приложений, которые отличаются устойчивым набором требований к задержкам и потерям пакетов, а также тем, что генерируют трафик с по-

стоянной (CBR) или переменной (VBR) битовой скоростью. Класс X зарезервирован для уникальных приложений, набор характеристик и требований которых не относится ни к одному из первых четырех классов.

Однако на какое количество классов мы бы ни разбивали существующий трафик, принципиальная задача от этого не меняется — нужно найти решение для успешного сосуществования в одном канале и эластичных, и чувствительных к задержкам классов трафика. Требования этих классов почти всегда противоречат друг другу. Одним из таких противоречий является требование к размеру кадра.

Эластичный трафик выигрывает от увеличения размера кадра, так как при этом снижаются накладные расходы на служебную информацию. Мы видели на примере Ethernet, что скорость передачи пользовательской информации может изменяться почти в два раза при изменении размера поля данных от его минимальной величины в 46 байт до максимальной в 1500 байт. Конечно, размер кадра не может увеличиваться до бесконечности, так как при этом теряется сама идея коммутации пакетов. Тем не менее для эластичного трафика при современном уровне скоростей размер кадра в несколько тысяч байтов является вполне приемлемым.

Напротив, чувствительный к задержкам трафик обслуживается лучше при использовании кадров небольшого размера в несколько десятков байтов. При применении больших кадров начинают проявляться два нежелательных эффекта:

- ожидание низкоприоритетных кадров в очередях;
- задержка пакетизации.

Рассмотрим эти эффекты на примере голосового трафика.

Мы знаем, что *время ожидания кадра в очереди* можно сократить, если обслуживать кадры чувствительного к задержкам трафика в приоритетной очереди. Однако если размер кадра может меняться в широком диапазоне, то даже при придании чувствительным к задержкам кадрам высшего приоритета обслуживания в коммутаторах время ожидания компьютерного пакета может все равно оказаться недопустимо высоким. Например, пакет в 4500 байт будет в течение 18 мс передаваться в выходной порт на скорости 2 Мбит/с (максимальная скорость работы порта коммутатора Frame Relay). При совмещении трафика за это время необходимо через тот же порт передать 144 замера голоса. Прерывать передачу пакета в сетях нежелательно, так как при распределенном характере сети накладные расходы на оповещение соседнего коммутатора о прерывании пакета, а потом — о возобновлении передачи пакета с прерванного места оказываются слишком большими.

Другой причиной явилось стремление ограничить еще одну составляющую задержки доставки данных — задержку пакетизации. **Задержка пакетизации** равна времени, в течение которого первый замер голоса ждет момента окончательного формирования пакета и отправки его по сети.

Механизм образования этой задержки иллюстрирует рис. 19.12.

На рисунке показан голосовой кодек — устройство, которое представляет голос в цифровой форме. Пусть он выполняет замеры голоса в соответствии со стандартной частотой 8 КГц (то есть через каждые 125 мкс), кодируя каждый замер одним байтом данных. Если мы используем для передачи голоса кадры Ethernet максимального размера, то в один кадр поместится 1500 замеров голоса. В результате первый замер, помещенный в кадр Ethernet, вынужден будет ждать отправки кадра в сеть $(1500 - 1) \times 125 = 187\,375$ мкс, или около 187 мс. Это весьма большая задержка для голосового трафика. Рекомендации стандартов

говорят о величине 150 мс как о максимально допустимой *суммарной* задержке голоса, в которую задержка пакетизации входит как одно из слагаемых.

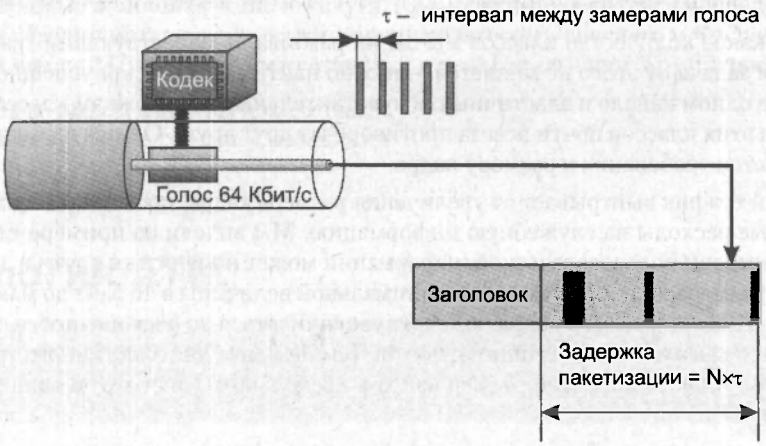


Рис. 19.12. Задержка пакетизации

ВНИМАНИЕ

Важно отметить, что задержка пакетизации не зависит от битовой скорости протокола, а зависит только от частоты работы кодека и размера поля данных кадра. Это отличает ее от задержки ожидания в очереди, которая снижается с возрастанием битовой скорости.

Размер ячейки ATM в 53 байта с полем данных 48 байт стал результатом компромисса между требованиями, предъявляемыми к сети при передаче эластичного и чувствительного к задержкам вариантов трафика. Можно сказать также, что компромисс был достигнут между телефонистами и компьютерщиками — первые настаивали на размере поля данных в 32 байта, а вторые — в 64 байта.

При размере поля данных в 48 байт одна ячейка ATM обычно переносит 48 замеров голоса, которые делаются с интервалом в 125 мкс. Поэтому первый замер должен ждать примерно 6 мс, прежде чем ячейка будет отправлена по сети. Именно по этой причине телефонисты боролись за уменьшения размера ячейки, так как 6 мс — это задержка, близкая к пределу, за которым начинаются нарушения качества передачи голоса. При выборе размера ячейки в 32 байта задержка пакетизации составила бы 4 мс, что гарантировало бы более качественную передачу голоса. А стремление компьютерных специалистов увеличить поле данных хотя бы до 64 байт вполне понятно — при этом повышается полезная скорость передачи данных. Избыточность служебных данных при использовании 48-байтного поля данных составляет 10 %, а при использовании 32-байтного поля данных она сразу повышается до 16 %.

Виртуальные каналы ATM

В сетях ATM поддерживается два типа виртуальных каналов:

- **постоянный виртуальный канал** (Permanent Virtual Circuit, PVC);
- **коммутируемый виртуальный канал** (Switched Virtual Circuit, SVC), создание такого канала происходит динамически по инициативе конечного узла с использованием автоматической процедуры.

Каналы PVC аналогичны каналам такого же типа в сетях Frame Relay, а для поддержки динамически устанавливаемых каналов SVC в технологии ATM добавлен специальный **протокол сигнализации** — это протокол, с помощью которого абоненты сети могут оперативно устанавливать каналы SVC. Такой тип протокола используется в телефонных сетях для установления соединения между телефонами абонентов. Для того чтобы протокол сигнализации мог работать, конечные узлы сети ATM получили глобально уникальные 20-разрядные адреса, иначе абонент, являющийся инициатором установления виртуального канала, не смог бы указать, с каким абонентом он хочет связаться.

В технологии ATM имеется также протокол маршрутизации PNNI (Private Network to Network Interface — интерфейс связи между частными сетями).

С целью обеспечения масштабируемости в сетях ATM введено два уровня иерархии виртуальных каналов: **виртуальный путь** (virtual path) и **виртуальное соединение** (virtual circuit). Виртуальный путь определяется старшей частью номера метки виртуального канала, а виртуальное соединение — младшей. Каждый виртуальный путь включает в себя до 4096 виртуальных соединений, проходящих внутри этого пути. Достаточно определить маршрут для пути, и все соединения, которые находятся внутри этого пути, будут ему следовать.

Категории услуг ATM

Для поддержания требуемого качества обслуживания и рационального расходования ресурсов в технологии ATM реализовано несколько служб. Услуги этих служб разбиты на категории, которые, в общем, соответствуют классам трафика, поступающим на вход сети.

Всего на уровне протокола ATM определено пять категорий услуг:

- **CBR** (Constant Bit Rate) — для трафика с постоянной битовой скоростью, например голосового;
- **rtVBR** (real-time Variable Bit Rate) — для трафика с переменной битовой скоростью, требующего соблюдения средней скорости передачи данных и синхронизации источника и приемника (примером является видеотрафик с переменной битовой скоростью, который вырабатывают многие видеокодеки за счет использования опорных кадров и кадров, описывающих изменения изображения относительно опорного кадра);
- **nrtVBR** (non real-time Variable Bit Rate) — для трафика с переменной битовой скоростью, требующего соблюдения средней скорости передачи данных и не требующего синхронизации источника и приемника;
- **ABR** (Available Bit Rate) — для трафика с переменной битовой скоростью, требующего соблюдения некоторой минимальной скорости передачи данных и не требующего синхронизации источника и приемника;
- **UBR** (Unspecified Bit Rate) — для трафика, не предъявляющего требований к скорости передачи данных и синхронизации источника и приемника.

Отсюда видно, что сети ATM отличаются от сетей Frame Relay большей степенью соответствия услуг требованиям трафика определенного типа, так как в сетях ATM нужный уровень обслуживания задается не только численными значениями параметров CIR, Bc и Be, но и самой категорией услуги.

Технология ATM, как и технология Frame Relay, пережила пик своей популярности, и сейчас область ее применения быстро сужается. Одной из причин этого стало появление сетей DWDM и расширение верхней границы скорости сетей Ethernet, предоставляющих

относительно дешевую пропускную способность. Еще одной причиной снижения интереса к ATM стала сложность этой технологии. В частности, некоторые проблемы возникают из-за использования ячеек маленького размера — на высоких скоростях оборудование с трудом справляется с обработкой таких интенсивных потоков ячеек (сравните количество кадров Ethernet максимальной длины с количеством ячеек ATM, необходимых для передачи одного и того же объема информации с той же самой скоростью).

Как и в случае Frame Relay, появление технологии MPLS, которая, с одной стороны, обладает некоторыми свойствами ATM, например поддерживает детерминированность маршрутов (это общее свойство технологий, основанных на технике виртуальных путей), а с другой — использует кадры любого формата и тесно интегрирована с IP, усугубило положение ATM. Одной из областей, где ATM по-прежнему удерживает позиции, является широкополосный доступ в Интернет. Если вы посмотрите на конфигурацию вашего домашнего маршрутизатора ADSL, то, скорее всего, увидите там записи, относящиеся к стеку ATM.

Более подробную информацию вы можете найти на сайте www.olifer.co.uk в разделе «Технология ATM».

Виртуальные частные сети

Услуга виртуальных частных сетей является одной из основных услуг, которую предоставляют сети FR и ATM. Вооруженные знанием основных принципов работы технологий FR и ATM, мы теперь можем более подробно рассмотреть и классифицировать эти услуги. Любая систематизация знаний полезна сама по себе, кроме того, она нам понадобится при изучении технологий MPLS и Carrier Ethernet, которые формировались во многом для реализации услуг VPN.

Из самого названия — **виртуальная частная сеть** — следует, что она каким-то образом воспроизводит свойства *реальной частной сети*. Без всяких натяжек назвать сеть *частной* можно только в том случае, если предприятие единолично владеет и управляет всей сетевой инфраструктурой — кабелями, кроссовым оборудованием, каналообразующей аппаратурой, коммутаторами, маршрутизаторами и другим коммуникационным оборудованием.

Главным отличием частной сети от общедоступной сети или сети, совместно используемой несколькими предприятиями, является ее *изолированность*.

Перечислим, в чем выражается эта изолированность.

- *Независимый выбор сетевых технологий.* Выбор ограничивается только возможностями производителей оборудования.
- *Независимая система адресации.* В частных сетях нет ограничений на выбор адресов — они могут быть любыми.
- *Предсказуемая производительность.* Собственные линии связи гарантируют заранее известную пропускную способность между узлами предприятия (для глобальных соединений) или коммуникационными устройствами (для локальных соединений).

- *Максимально возможная безопасность*. Отсутствие связей с внешним миром ограждает сеть от атак извне и существенно снижает вероятность «прослушивания» трафика по пути следования.

Однако частная сеть — решение крайне неэкономичное! Такие сети, особенно в национальном или международном масштабах, могут себе позволить только очень крупные и богатые предприятия. Создание частной сети — привилегия тех, кто имеет производственные предпосылки для разработки собственной сетевой инфраструктуры. Например, нефтяные или газовые компании способны с относительно невысокими издержками прокладывать собственные технологические кабели связи вдоль трубопроводов. Частные сети были популярны в относительно далеком прошлом, когда общедоступные сети передачи данных были развиты очень слабо. Сегодня же их почти повсеместно вытеснили сети VPN, которые представляют собой компромисс между качеством услуг и их стоимостью.

В зависимости от того, кто реализует сети VPN, они подразделяются на два вида.

- **Поддерживаемая клиентом виртуальная частная сеть** (Customer Provided Virtual Private Network, CPVPN) отражает тот факт, что все тяготы по поддержке сети VPN ложатся на плечи потребителя. Поставщик предоставляет только «простые» традиционные услуги общедоступной сети по объединению узлов клиента, а специалисты предприятия самостоятельно конфигурируют средства VPN и управляют ими.
- В случае **поддерживаемой поставщиком виртуальной частной сети** (Provider Provisioned Virtual Private Network, PPVPN) поставщик услуг на основе собственной сети воспроизводит частную сеть для каждого своего клиента, изолируя и защищая ее от остальных. Такой способ организации VPN сравнительно нов и не столь широко распространен, как первый.

В последние год-два популярность сетей PPVPN растет — заботы по созданию и управлению VPN довольно обременительны и специфичны, поэтому многие предприятия предпочитают переложить их на плечи надежного поставщика. Реализация услуг VPN позволяет поставщику оказывать и ряд дополнительных услуг, включая контроль за работой клиентской сети, веб-хостинг и хостинг почтовых служб, хостинг специализированных приложений клиентов.

Помимо деления сетей VPN на CPVPN и PPVPN существует еще и другая классификация — в зависимости от места расположения устройств, выполняющих функции VPN. Виртуальная частная сеть может строиться:

- **на базе оборудования, установленного на территории потребителя** (Customer Premises Equipment based VPN, CPE-based VPN, или Customer Edge based VPN, CE-based VPN);
- **на базе собственной инфраструктуры поставщика** (Network-based VPN, или Provider Edge based VPN, PE-based VPN).

В любом случае основную часть функций (или даже все) по поддержанию VPN выполняют пограничные устройства сети — либо потребителя, либо поставщика.

Сети, поддерживаемые поставщиком, могут строиться как на базе инфраструктуры поставщика, так и на базе оборудования, установленного на территории потребителя. Первый вариант наиболее понятен: поставщик управляет расположенным в его сети оборудованием. Во втором случае оборудование VPN расположено на территории клиента, но поставщик управляет им удаленно, что освобождает специалистов предприятия-клиента от достаточно сложных и специфических обязанностей.

Когда VPN поддерживается клиентом (CPVPN), оборудование всегда находится в его сети, то есть VPN строится на базе устройств клиента (CE-based).

Сеть VPN, как и любая *имитирующая система*[1], характеризуется, во-первых, тем, какие свойства объекта имитируются, во-вторых, степенью приближенности к оригиналу, в-третьих, используемыми средствами имитации.

Рассмотрим, какие элементы частной сети являются предметом «виртуализации» в VPN.

Практически все сети VPN имитируют *собственные каналы* в сетевой инфраструктуре поставщика, предназначенной для обслуживания множества клиентов.

В том случае, когда имитируется инфраструктура каналов одного предприятия, то услуги VPN называют также услугами **интранет** (intranet), или **внутренней сети**, а в том случае, когда к таким каналам добавляются также каналы, соединяющие предприятие с его предприятиями-партнерами, с которыми также необходимо обмениваться информацией в защищенном режиме, — услугами **экстранет** (extranet), или **внешней сети**.

Термин «виртуальная частная сеть» применяется только тогда, когда «собственные» физические каналы имитируются средствами пакетных технологий: ATM, Frame Relay, IP, IP/MPLS или Carrier Ethernet. Качество связи между узлами клиентов в этом случае уже вполне ощутимо отличается от того, которое было бы при их реальном соединении собственным физическим каналом. В частности, появляется неопределенность пропускной способности и других характеристик связи, поэтому определение «виртуальная» становится здесь уместным. При применении пакетных сетей для построения VPN клиентам предоставляются не только физические каналы, но и определенная технология канального уровня (например, ATM или Frame Relay), а при использовании IP — и сетевого.

Виртуальная частная сеть может имитировать не только физические каналы, но и более высокоуровневые свойства сети. Так, может быть спроектирована сеть VPN, способная поддерживать IP-трафик клиента с созданием эффекта изолированной IP-сети. В этом случае VPN производит некоторые дополнительные сетевые операции над клиентским трафиком — сбор разнообразной статистики, фильтрацию и экранирование взаимодействий между пользователями и подразделениями одного и того же предприятия (не нужно путать с экранированием от внешних пользователей — это основная функция VPN) и т. п.

Имитация сервисов прикладного уровня встречается в VPN гораздо реже, чем имитация собственно транспортных функций, но также возможна. Например, поставщик в состоянии поддерживать для клиента веб-сайты, почтовую систему или специализированные приложения управления предприятием.

Другим критерием, используемым при сравнении VPN, является *степень приближенности сервисов, предлагаемых VPN, к свойствам сервисов частной сети.*

Во-первых, важнейшим свойством сервисов частной сети является *безопасность*. Безопасность VPN подразумевает весь набор атрибутов защищенной сети — конфиденциальность, целостность и доступность информации при передаче через общедоступную сеть, а также защищенность внутренних ресурсов сетей потребителя и поставщика от внешних атак. Степень безопасности VPN варьируется в широких пределах в зависимости от применяемых средств защиты: шифрования трафика, аутентификации пользователей и устройств изоляции адресных пространств (например, на основе техники NAT), использования виртуальных каналов и двухточечных туннелей, затрудняющих подключение к ним

[1] В данном случае VPN рассматривается как имитация частной сети предприятия.

несанкционированных пользователей. Так как ни один способ защиты не дает абсолютных гарантий, то средства безопасности могут комбинироваться для создания эшелонированной обороны.

Во-вторых, желательно, чтобы сервисы VPN приближались к сервисам частной сети *по качеству обслуживания*. Качество транспортного обслуживания подразумевает, в первую очередь, гарантии пропускной способности для трафика клиента, к которым могут добавляться и другие параметры QoS — максимальные задержки и процент потерянных данных. В пакетных сетях пульсации трафика, переменные задержки и потери пакетов — неизбежное зло, поэтому степень приближения виртуальных каналов к каналам TDM всегда неполная и вероятностная (в среднем, но никаких гарантий для отдельно взятого пакета). Разные пакетные технологии отличаются различным уровнем поддержки параметров QoS. В ATM, например, механизмы качества обслуживания наиболее совершенны и отработаны, а в IP-сетях они только начинают внедряться. Поэтому далеко не каждая сеть VPN пытается воссоздать эти особенности частной сети. Считается, что безопасность — обязательное свойство VPN, а качество транспортного обслуживания — только желательное.

В-третьих, сеть VPN приближается к реальной частной сети, если она обеспечивает для клиента *независимость адресного пространства*. Это дает клиенту одновременно и удобство конфигурирования, и способ поддержания безопасности. Причем желательно, чтобы не только клиенты ничего не знали об адресных пространствах друг друга, но и магистраль поставщика имела собственное адресное пространство, неизвестное пользователям. В этом случае сеть поставщика услуг будет надежнее защищена от умышленных атак или неумышленных действий своих клиентов, а значит, более высоким будет качество предоставляемых услуг VPN.

Существенное влияние на свойства виртуальных частных сетей оказывают технологии, с помощью которых эти сети строятся. Все технологии VPN можно разделить на два класса в зависимости от того, каким образом они обеспечивают безопасность передачи данных:

- технологии разграничения трафика;
- технологии шифрования.

Сети VPN на основе техники шифрования рассматриваются в главе 24.

В технологиях разграничения трафика используется техника постоянных виртуальных каналов, обеспечивающая надежную защиту трафика каждого клиента от намеренного или ненамеренного доступа к нему других клиентов публичной сети. К этому типу технологий относятся:

- ATM VPN;
- Frame Relay VPN;
- MPLS VPN;
- Carrier Ethernet VPN.

Двухточечные виртуальные каналы этих технологий имитируют сервис выделенных каналов, проходя от пограничного устройства (Client Edge, CE) одного сайта клиента через поставщика к CE другого сайта клиента.

ВНИМАНИЕ

Под термином «сайт» здесь понимается территориально обособленный фрагмент сети клиента. Например, о корпоративной сети, в которой сеть центрального отделения связывается с тремя удаленными филиалами, можно сказать, что она состоит из четырех сайтов.

Защита данных достигается благодаря тому, что несанкционированный пользователь не может подключиться к постоянному виртуальному каналу, не изменив таблицы коммутации устройств поставщика услуг, а значит, ему не удастся провести атаку или прочитать данные. Свойство защищенности трафика является *естественным свойством* техники виртуальных каналов, поэтому сервисы ATM VPN и Frame Relay VPN являются на самом деле не чем иным, как обычными сервисами PVC сетей ATM или Frame Relay. Любой пользователь ATM или Frame Relay, использующий инфраструктуру PVC для связи своих локальных сетей, потребляет услугу VPN даже в том случае, когда он это явно не осознает. Это одно из «родовых» преимуществ техники виртуальных каналов по сравнению с дейтаграммной техникой, так как при применении последней без дополнительных средств VPN пользователь оказывается не защищенным от атак любого другого пользователя сети.

Так как в технологиях ATM и Frame Relay при передаче данных используются только два уровня стека протоколов, варианты VPN, построенные на их основе, называют также **сетями VPN уровня 2** (Layer 2 VPN, L2VPN). Наличие в технологиях ATM и Frame Relay механизмов поддержания параметров QoS позволяет ATM VPN и Frame Relay VPN достаточно хорошо приближаться к частным сетям на выделенных каналах.

Информация третьего уровня никогда не анализируется и не меняется в этих сетях — это одновременно и достоинство, и недостаток. Преимущество в том, что клиент может передавать по такому виртуальному каналу трафик любых протоколов, а не только IP. Кроме того, IP-адреса клиентов и поставщика услуг изолированы и независимы друг от друга — они могут выбираться произвольным образом, так как не используются при передаче трафика через магистраль поставщика. Никаких других знаний о сети поставщика услуг, помимо значений меток виртуальных каналов, клиенту не требуется. Недостаток этого подхода состоит в том, что поставщик не оперирует IP-трафиком клиента и, следовательно, не может оказывать дополнительные услуги, связанные с сервисами IP, а это сегодня очень перспективное направление бизнеса поставщиков услуг.

Главным недостатком сети L2VPN является ее сложность и достаточно высокая стоимость. При организации полносвязной топологии сайтов клиента зависимость операций конфигурирования от числа сайтов имеет квадратичный характер (рис. 19.13, *а*).

Действительно, для соединения N сайтов необходимо создать $N \times (N-1)/2$ двунаправленных виртуальных каналов или $N \times (N-1)$ однонаправленных. В частности, при значении N, равном 100, потребуется 5000 операций конфигурирования. И хотя они и выполняются с помощью автоматизированных систем администрирования, ручной труд и вероятность ошибки все равно сохраняются. При поддержке только услуг *интранет* общее количество конфигурируемых соединений прямо пропорционально количеству клиентов — и это хорошо! Но оказание услуг *экстранет* ухудшает ситуацию, так как подразумевает необходимость обеспечить связь сайтов разных клиентов. Масштабируемость сети ATM/FR VPN можно улучшить, если клиент откажется от полносвязной топологии и организует связи типа «звезда» через один или несколько выделенных транзитных сайтов (рис. 19.13, *б*). Конечно, производительность сети клиента при этом снизится, так как увеличится число транзитных передач информации. Однако экономия средств будет налицо — поставщики услуг взимают деньги за свои виртуальные каналы, как правило, «поштучно».

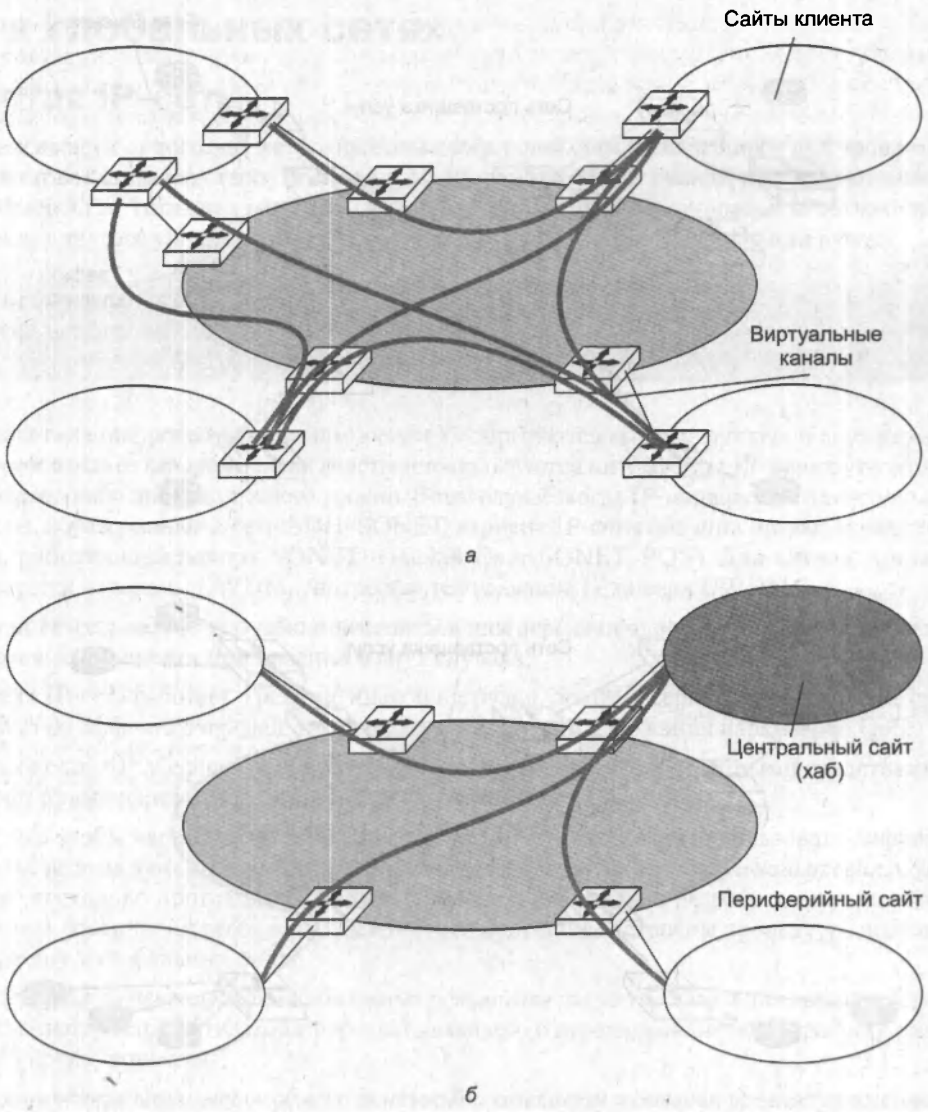


Рис. 19.13. Масштабируемость сети L2VPN

Клиенты сети ATM/FR VPN не могут нанести ущерб друг другу, а также атаковать IP-сеть поставщика. Сегодня поставщик услуг всегда располагает IP-сетью, даже если он оказывает только услуги ATM/FR VPN. Без IP-сети и ее сервисов администрирования он просто не сможет управлять своей сетью ATM/FR. IP-сеть является оверлейной (наложенной) по отношению к сетям ATM или FR, поэтому клиенты ATM/FR ничего не знают о ее структуре и даже о ее наличии (рис. 19.14).

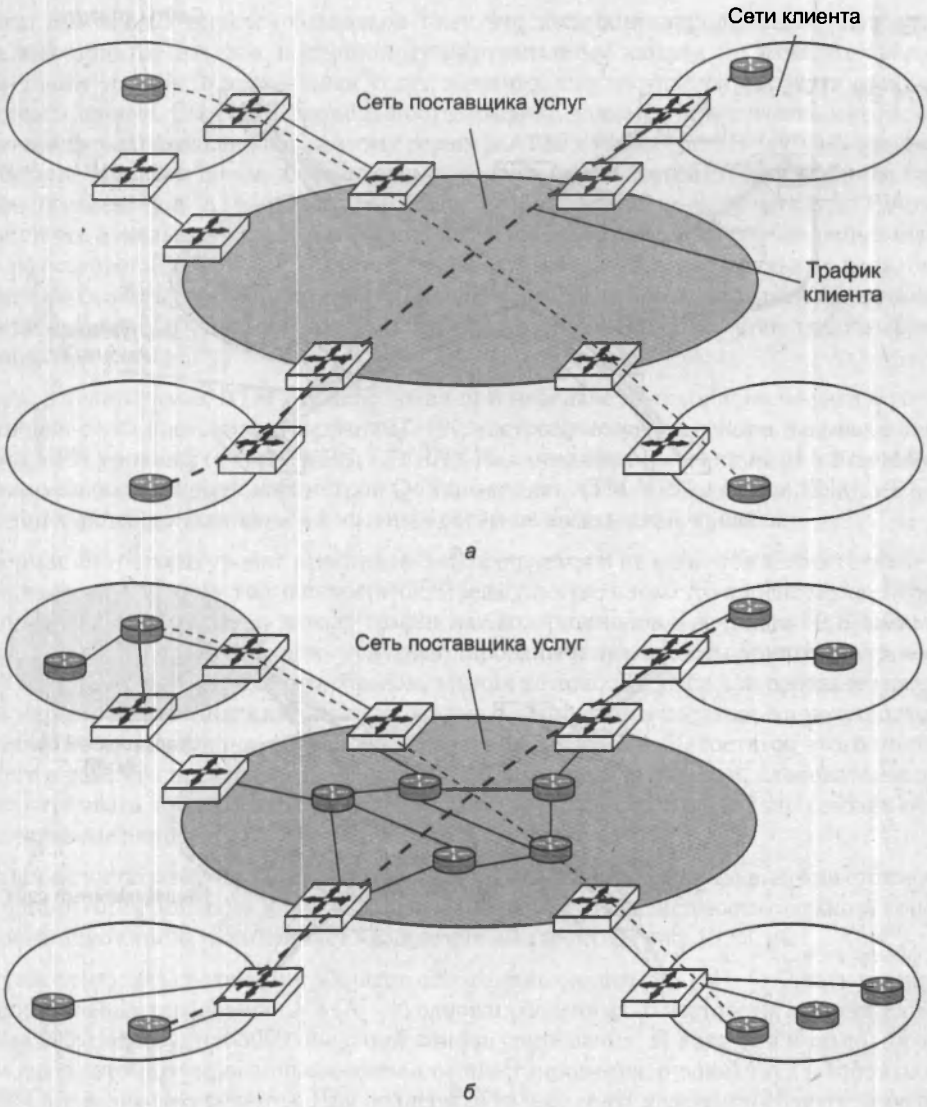


Рис. 19.14. Оверлейная (*а*) и одноранговая (*б*) модели VPN

Сети MPLS VPN могут строиться как по схеме L2VPN, так и по другой схеме, использующей протоколы трех уровней. Такие сети называют **сетями VPN уровня 3** (Layer 3 VPN, L3VPN). В технологии L3VPN также применяется техника LSP для разграничения трафика клиентов внутри сети поставщика услуг, поддерживающей технологию MPLS.

Сеть L3VPN взаимодействует с сетями клиентов на основе IP-адресов, а L2VPN — на основе адресной информации второго уровня, например MAC-адресов или идентификаторов виртуальных каналов Frame Relay.

IP в глобальных сетях

Чистая IP-сеть

В зависимости от того, как устроены слои глобальной сети, находящиеся под уровнем IP, можно говорить о «чистых» IP-сетях и об IP «поверх» (over) какой-нибудь технологии, например ATM. Название «чистая» IP-сеть говорит о том, что под уровнем IP не находится никакого другого уровня, выполняющего коммутацию пакетов (кадров или ячеек).

Чистая IP-сеть отличается от многослойной тем, что под уровнем IP нет другой сети с коммутацией пакетов, такой как ATM или Frame Relay, и IP-маршрутизаторы связываются между собой выделенными каналами (физическими или соединениями PDH/SDH/DWDM).

В такой сети цифровые каналы по-прежнему образуются инфраструктурой двух нижних уровней, а этими каналами непосредственно пользуются интерфейсы IP-маршрутизаторов без какого-либо промежуточного уровня. В том случае, когда IP-маршрутизатор использует каналы, образованные в сети SDH/SONET, вариант IP-сети получил название **пакетной сети, работающей поверх SONET**[1] (Packet Over SONET, POS). Для случая, когда IP пользуется каналами DWDM, употребляется название IP поверх DWDM.

Чистая IP-сеть может успешно применяться для передачи чувствительного к задержкам трафика современных приложений в двух случаях:

- если IP-сеть работает в режиме низкой нагрузки, поэтому сервисы всех типов не страдают от эффекта очередей, так что сеть не требует поддержания параметров QoS;
- если слой IP обеспечивает поддержку параметров QoS собственными средствами за счет применения механизмов IntServ или DiffServ.

Для того чтобы маршрутизаторы в модели чистой IP-сети могли использовать цифровые каналы, на этих каналах *должен работать какой-либо протокол канального уровня*. Существует несколько протоколов канального уровня, специально разработанных для двухточечных соединений глобальных сетей. В эти протоколы встроены процедуры, полезные при работе в глобальных сетях:

- *взаимная аутентификация удаленных устройств* часто требуется для защиты сети от «ложного» маршрутизатора, перехватывающего и перенаправляющего трафик с целью его прослушивания;
- *согласование параметров обмена данными на канальном и сетевом уровнях* применяется при удаленном взаимодействии, когда два устройства расположены в разных городах, перед началом обмена часто необходимо автоматически согласовывать такие, например, параметры, как MTU.

Из набора существующих двухточечных протоколов протокол IP сегодня использует два: HDLC и PPP. Существует также устаревший протокол SLIP (Serial Line Internet Protocol — межсетевой протокол для последовательного канала), который долгое время был основным протоколом удаленного доступа индивидуальных клиентов к IP-сети через телефонную сеть. Однако сегодня он уже не применяется.

---

[1] Название международной версии SDH было опущено разработчиками технологии POS.

Помимо уже упомянутых протоколов, в глобальных сетях на выделенных каналах IP-маршрутизаторы нередко используют какой-либо из высокоскоростных вариантов Ethernet: Fast Ethernet, Gigabit Ethernet или 10G Ethernet. Усовершенствования, сделанные в технологии Carrier Ethernet и направленные на повышение эксплуатационных свойств классического варианта Ethernet, отражают потребности применения этой технологии в глобальных сетях.

Протокол HDLC

Протокол HDLC (High-level Data Link Control — высокоуровневое управление линией связи), представляет целое семейство протоколов, реализующих функции канального уровня.

Первое, что мы отметим по поводу протокола HDLC, — его *функциональное разнообразие*. Он может работать в нескольких весьма отличающихся друг от друга режимах, поддерживает не только двухточечные соединения, но и соединения с одним источником и несколькими приемниками, он также предусматривает различные функциональные роли взаимодействующих станций. Сложность HDLC объясняется тем, что это очень «старый» протокол, разработанный еще в 70-е годы для ненадежных каналов связи. Поэтому в одном из режимов протокол HDLC подобно протоколу TCP поддерживает процедуру установления логического соединения и процедуры контроля передачи кадров, а также восстанавливает утерянные или поврежденные кадры. Существует и дейтаграммный режим работы HDLC, в котором логическое соединение не устанавливается, а кадры не восстанавливаются.

В IP-маршрутизаторах чаще всего используется версия протокола HDLC, разработанная компанией Cisco. Несмотря на то что эта версия является фирменным протоколом, она стала стандартом де-факто для IP-маршрутизаторов большинства производителей. Версия Cisco HDLC работает только в дейтаграммном режиме, что соответствует современной ситуации с незашумленными надежными каналами связи. По сравнению со стандартным протоколом версия Cisco HDLC включает несколько расширений, главным из которых является многопротокольная поддержка. Это означает, что в заголовок кадра Cisco HDLC добавлено поле типа протокола, подобное полю EtherType. Это поле содержит код протокола, данные которого переносит кадр Cisco HDLC. В стандартной версии HDLC такое поле отсутствует.

Протокол PPP

Протокол PPP (Point-to-Point Protocol — протокол двухточечной связи) является стандартным протоколом Интернета. Протокол PPP, так же как и HDLC, представляет собой целое семейство протоколов, в которое, в частности, входят:

- **протокол управления линией связи** (Link Control Protocol, LCP);
- **протокол управления сетью** (Network Control Protocol, NCP);
- **многоканальный протокол PPP** (Multi Link PPP, MLPPP);
- **протокол аутентификации по паролю** (Password Authentication Protocol, PAP);
- **протокол аутентификации по квитированию вызова** (Challenge Handshake Authentication Protocol, CHAP).

ПРИМЕЧАНИЕ

При разработке протокола PPP за основу был взят формат кадров HDLC и дополнен несколькими полями. Эти дополнительные поля протокола PPP вложены в поле данных кадра HDLC. Позже были разработаны стандарты, описывающие вложение кадра PPP в кадры Frame Relay и других протоколов глобальных сетей. Хотя протокол PPP и работает с кадром HDLC, он не поддерживает, подобно стандартной версии протокола HDLC, процедуры надежной передачи кадров и управления их потоком.

Особенностью протокола PPP, отличающей его от других протоколов канального уровня, является *сложная переговорная процедура* принятия параметров соединения. Стороны обмениваются различными параметрами, такими как качество линии, размер кадров, тип протокола аутентификации и тип инкапсулируемых протоколов сетевого уровня.

В корпоративной сети конечные системы часто отличаются размерами буферов для временного хранения пакетов, ограничениями на размер пакета, списком поддерживаемых протоколов сетевого уровня. Физическая линия, связывающая конечные устройства, может варьироваться от низкоскоростной аналоговой до высокоскоростной цифровой линии с различными уровнями качества обслуживания.

Протокол, в соответствии с которым принимаются параметры соединения, называется *протоколом управления линией связи* (LCP). Чтобы справиться со всеми возможными ситуациями, в протоколе PPP имеется набор стандартных параметров, действующих по умолчанию и учитывающих все стандартные конфигурации. При установлении соединения два взаимодействующих устройства для нахождения взаимопонимания пытаются сначала использовать эти параметры. Каждый конечный узел описывает свои возможности и требования. Затем на основании этой информации принимаются параметры соединения, устраивающие обе стороны. Переговорная процедура протоколов может и не завершиться соглашением о каком-нибудь параметре. Если, например, один узел предлагает в качестве MTU значение 1000 байт, а другой отвергает это предложение и в свою очередь предлагает значение 1500 байт, которое отвергается первым узлом, то по истечении тайм-аута переговорная процедура может закончиться безрезультатно.

Одним из важных параметров соединения PPP является *режим аутентификации*. Для целей аутентификации PPP предлагает по умолчанию *протокол аутентификации по паролю* (PAP), передающий пароль по линии связи в открытом виде, или *протокол аутентификации по квитированию вызова* (CHAP), не передающий пароль по линии связи и поэтому обеспечивающий более высокий уровень безопасности сети. Пользователям также разрешается добавлять новые алгоритмы аутентификации. Кроме того, пользователи могут влиять на выбор алгоритмов сжатия заголовка и данных.

Многопротокольная поддержка — способность протокола PPP поддерживать несколько протоколов сетевого уровня — обусловила распространение PPP как стандарта де-факто. Внутри одного соединения PPP могут передаваться потоки данных различных сетевых протоколов, включая IP, Novell IPX и многих других, сегодня уже не употребляющихся, а также данные протоколов канального уровня локальной сети.

Каждый протокол сетевого уровня конфигурируется отдельно с помощью соответствующего *протокола управления сетью* (NCP). Под конфигурированием понимается, во-первых, констатация того факта, что данный протокол будет использоваться в текущем сеансе PPP, а во-вторых, переговорное согласование некоторых параметров протокола. Больше всего параметров устанавливается для протокола IP, включая IP-адреса взаимодействующих

узлов, IP-адреса DNS-серверов, признак компрессии заголовка IP-пакета и т. д. Для каждого протокола конфигурирования протокола верхнего уровня, помимо общего названия NCP, употребляется особое название, построенное путем добавления аббревиатуры CP (Control Protocol — протокол управления) к имени конфигурируемого протокола, например для IP — это протокол IPCP, для IPX — IPXCP и т. п.

Расширяемость протокола. Под этим свойством PPP понимается как возможность включения новых протоколов в стек PPP, так и возможность применения собственных протоколов пользователей вместо рекомендуемых в PPP по умолчанию. Это позволяет наилучшим образом настроить PPP для каждой конкретной ситуации.

Одной из привлекательных способностей протокола PPP является способность использования нескольких физических линий связи для образования одного логического канала, то есть агрегирование каналов (об агрегировании линий связи см. также в главе 14). Эту возможность реализует *многоканальный протокол PPP* (MLPPP).

Использование выделенных линий IP-маршрутизаторами

Схема использования выделенной линии маршрутизатором показана на рис. 19.15. Для соединения порта маршрутизатора с выделенной линией необходимо устройство DCE соответствующего типа. Это устройство призвано обеспечить согласование физического интерфейса маршрутизатора с интерфейсом физического уровня, используемого выделенной линией, например V.35 с T1.

Рис. 19.15. Соединение IP-сетей с помощью выделенной линии

Если выделенная линия является аналоговой, то устройством DCE будет модем, а если цифровой — то устройство DSU/CSU.

Порт маршрутизатора может включать встроенное устройство DCE. Например, маршрутизатор, рассчитанный на работу с каналом SDH, обычно имеет встроенный порт с интерфейсом SDH определенной скорости STM-N.

Встроенные порты PDH/SDH могут как поддерживать, так и не поддерживать внутреннюю структуру кадров этих технологий. В том случае, когда порт различает подкадры, из которых состоит кадр, например отдельные тайм-слоты кадра E1 или отдельные виртуаль-

ные контейнеры VC-12 (2 Мбит/с), входящие в кадр STM-1, и порт может использовать их как отдельные физические подканалы, то говорят, что это **порт с разделением каналов**. Каждому такому каналу присваивается отдельный IP-адрес. В противном случае порт целиком рассматривается как один физический канал с одним IP-адресом.

В качестве примера на рис. 19.15 выбрано соединение двух маршрутизаторов через цифровой канал E1, установленный в сети PDH. Маршрутизатор использует для подключения к каналу устройство DSU/CSU с внутренним интерфейсом RS-449 и внешним интерфейсом G.703, который определен в качестве интерфейса доступа к каналам PDH.

Маршрутизаторы после подключения к выделенной линии и локальной сети необходимо конфигурировать. Выделенный канал является отдельной IP-подсетью, как и локальные подсети 1 и 2, которые он соединяет. Этой подсети можно также дать некоторый IP-адрес из диапазона адресов, которым распоряжается администратор составной сети. В приведенном примере выделенному каналу присвоен адрес подсети 201.20.23.64, состоящей из двух узлов, что определяется маской 255.255.255.252.

Интерфейсам маршрутизаторов, связанных выделенной линией, можно и не присваивать IP-адрес — такой интерфейс маршрутизатора называется **ненумерованным**. Действительно, отсылая пакеты протокола маршрутизации (RIP или OSPF) по выделенному каналу, маршрутизаторы непременно их получат. Протокол ARP на выделенном канале не используется, так как аппаратные адреса на таком канале не имеют практического смысла.

Работа IP-сети поверх сети ATM

Рассмотрим взаимодействие слоя IP со слоем ATM на примере сети, представленной на рис. 19.16.

В сети ATM проложено шесть постоянных виртуальных каналов, соединяющих порты IP-маршрутизаторов. Каждый порт маршрутизатора в качестве конечного узла должен поддерживать технологию ATM. После того как виртуальные каналы установлены, маршрутизаторы могут пользоваться ими как физическими, посылая данные порту соседнего (по отношению к виртуальному каналу) маршрутизатора.

В сети ATM образуется сеть виртуальных каналов с собственной топологией. Топология виртуальных каналов, соответствующая сети, представленной на рис. 19.16, показана на рис. 19.17. Сеть ATM прозрачна для IP-маршрутизаторов, они ничего не знают о физических связях между портами коммутаторов ATM. IP-сеть является наложенной (оверлейной) по отношению к сети ATM.

Для того чтобы протокол IP мог корректно работать, ему необходимо знать соответствие между IP-адресами соседей и адресами виртуальных каналов ATM, с помощью которых достижим соответствующий IP-адрес. То есть нужно уметь отображать сетевые адреса на аппаратные, роль которых в данном случае играют адреса виртуальных каналов ATM. Другими словами, протоколу IP необходим некий вариант протокола ARP. Поскольку сеть ATM не поддерживает широковещательных запросов, таблица соответствия адресов не может быть создана автоматически. Администратор IP-сети должен вручную выполнить конфигурирование каждого интерфейса маршрутизатора, задав таблицу соответствия для всех номеров виртуальных каналов, исходящих из этого интерфейса и входящих в него. При этом физический интерфейс может быть представлен в виде набора логических интерфейсов (или *подинтерфейсов*), имеющих IP-адреса.

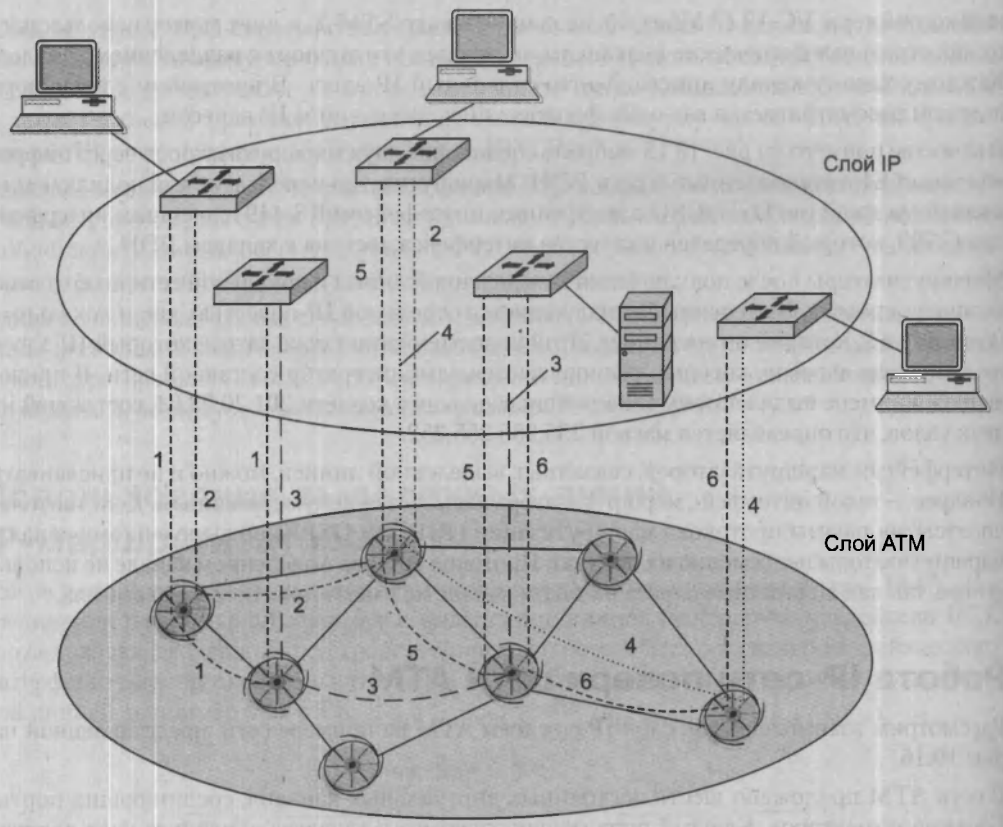


Рис. 19.16. Взаимодействие слоев IP и ATM

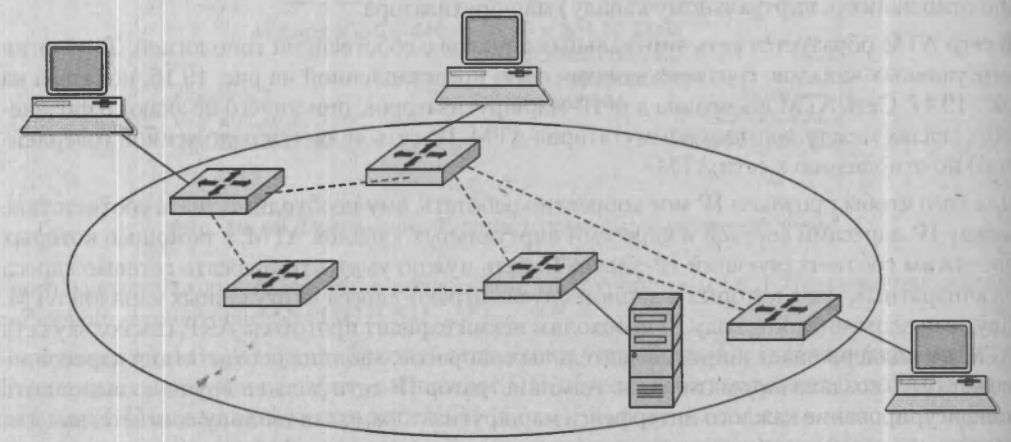


Рис. 19.17. Топология связей между маршрутизаторами

Например, в маршрутизаторах компании Cisco Systems команды конфигурирования логического интерфейса, соответствующего виртуальному каналу с адресом VPI/VCI, равным 0/36, выглядят следующим образом:

```
pvc 0/36
protocol ip 10.2.1.1
```

После выполнения этих команд маршрутизатор будет знать, что для пересылки пакета по адресу 10.2.1.1 ему потребуется разбить пакет на последовательность ячеек ATM (с помощью функции SAR интерфейса ATM) и отправить их все по постоянному виртуальному каналу с адресом 0/36.

Если многослойная сеть IP/ATM предназначается для передачи трафика различных классов с соблюдением параметров QoS для каждого класса, то соседние маршрутизаторы должны быть связаны несколькими виртуальными каналами, по одному для каждого класса. Маршрутизатору должна быть задана политика классификации пакетов, позволяющая отнести передаваемый пакет к определенному классу. Пакеты каждого класса направляются на соответствующий виртуальный канал, который обеспечивает трафику требуемые параметры QoS. Однако предварительно необходимо провести инжиниринг трафика для сети ATM, определив оптимальные пути прохождения трафика и соответствующим образом проложив виртуальные каналы. Результатом такой работы будет соблюдение требований к средним скоростям потоков, а коэффициент загрузки каждого интерфейса коммутаторов ATM не превысит определенной пороговой величины, гарантирующей каждому классу трафика приемлемый уровень задержек.

Выводы

Основными типами транспортных услуг глобальных компьютерных сетей являются услуги выделенных линий, доступа в Интернет и виртуальных частных сетей (VPN).

Сервис виртуальных частных сетей может быть реализован различными способами и с различной степенью приближения к сервису частных сетей на выделенных каналах, который он эмулирует.

Большинство современных глобальных сетей являются составными IP-сетями, а отличия между ними заключаются в технологиях, лежащих под уровнем IP.

Крупные глобальные сети часто строятся по четырехуровневой схеме, где два нижних уровня — это уровни первичной сети, образуемые технологиями DWDM и OTN/SDH. На основе первичной сети оператор сети строит каналы наложенной (оверлейной) сети — пакетной или телефонной. IP-сеть образует верхний уровень.

Каждый слой такой сети может выполнять две функции:

- предоставление услуг конечным пользователям;
- поддержка функций вышележащих уровней сети оператора.

Техника виртуальных каналов дает оператору сети большую степень контроля над путями прохождения данных, чем техника дейтаграммной передачи данных, применяемая в таких технологиях, как IP и Ethernet. По этой причине в большинстве технологий канального уровня, разработанных специально для глобальных сетей, таких как Frame Relay и ATM, используется техника виртуальных каналов.

Сети Frame Relay работают на основе постоянных виртуальных каналов. Эти сети создавались специально для передачи пульсирующего компьютерного трафика, поэтому при резервировании пропускной способности указывается средняя скорость передачи (CIR) и согласованный объем пульсаций (Bc).

Технология ATM является дальнейшим развитием идей предварительного резервирования пропускной способности виртуального канала, реализованных в технологии Frame Relay. Технология ATM поддерживает основные виды трафика для абонентов разного типа: трафик CBR, характерный для телефонных сетей и сетей передачи изображения, трафик VBR, характерный для компьютерных сетей, а также для передачи компрессированных голоса и изображения.

«Чистая» IP-сеть отличается от многослойной тем, что под уровнем IP нет другой сети с коммутацией пакетов, такой как ATM или Frame Relay, и IP-маршрутизаторы связываются между собой выделенными каналами (физическими или соединениями PDH/SDH/DWDM).

Из набора существующих двухточечных протоколов протокол IP сегодня использует два: HDLC и PPP. Каждый из них представляет целое семейство протоколов, работающих на канальном уровне.

Сеть VPN может быть реализована как самим предприятием, так и поставщиком услуг. Она может строиться на базе оборудования, установленного на территории и потребителя, и поставщика услуг.

Технологии VPN можно разделить на два класса в зависимости от того, каким образом они обеспечивают безопасность передачи данных: технологии разграничения трафика (ATM VPN, Frame Relay VPN, MPLS VPN) и технологии на основе шифрования (IPSec VPN).

Вопросы и задания

1. В чем заключаются преимущества услуг виртуальных частных сетей по сравнению с услугами выделенных каналов с точки зрения поставщика этих услуг? Варианты ответов:
   а) их легче конфигурировать;
   б) можно обслужить большее число клиентов, имея ту же инфраструктуру физических каналов связи;
   в) легче контролировать соглашения SLA.
2. В чем заключаются недостатки услуг виртуальных частных сетей по сравнению с услугами выделенных каналов с точки зрения клиентов? Варианты ответов:
   а) возможны задержки и потери пакетов;
   б) не всегда есть гарантии пропускной способности соединений;
   в) высокая стоимость услуг.
3. Причинами популярности техники виртуальных каналов в глобальных сетях являются следующие их свойства:
   а) высокая надежность;
   б) контроль над путями прохождения трафика;
   в) эффективность при оказании услуг VPN;
   г) эффективность работы по схеме «каждый с каждым».
4. В каких из приведенных примеров применяется туннелирование? Варианты ответов:
   а) передача IP-пакетов через сеть Frame Relay;
   б) передача кадров Ethernet с сохранением MAC-адресов через IP-сеть;
   в) передача зашифрованных IP-пакетов через Интернет.
5. Какой протокол чаще всего исполняет роль несущего протокола при туннелировании?

6. Уникальность метки DLCI должна быть обеспечена в пределах:

   а) сети Framew Relay данного провайдера;

   б) порта отдельного коммутатора сети;

   в) отдельного коммутатора сети.

7. В соглашении SLA между клиентом и поставщиком услуг Frame Relay оговаривается значение CIR = 512 Кбит/с на периоде 100 мс, при этом при подсчете скорости учитывается только поле данных кадров Frame Relay. На очередном периоде 100 мс пограничный коммутатор клиента послал в сеть 7 кадров с размерами поля данных 1000, 1500, 1200, 1500, 1000, 1300 и 1500 байт соответственно. Были ли эти кадры помечены пограничным коммутатором провайдера признаком DE = 1, а если да, то какие?

8. Какую категорию услуг целесообразно выбрать для передачи голоса через сеть ATM? Варианты ответов:

   а) CBR; б) rtVBR; б) ABR.

9. Задержка пакетизации это:

   а) время передачи пакета в линию связи;

   б) время между помещением в пакет первого и последнего замеров голоса;

   в) время ожидания пакета в очереди к выходному интерфейсу.

10. Избыточность служебных данных для ячеек ATM составляет:

    а) 8 %; б) 16 % в) 10 %.

11. Что отличает виртуальные каналы ATM от виртуальных каналов Frame Relay? Варианты ответов:

    а) двухуровневая иерархия;

    б) протокол маршрутизации PNNI;

    в) поддержка режима SVC.

12. Какие свойства частной сети имитирует услуга виртуальных частных сетей, предоставляемая провайдером? Варианты ответов:

    а) независимость адресных пространств;

    б) высокое качество обслуживания;

    в) защищенность передаваемых данных;

    г) независимость администрирования.

13. Чем отличаются услуги L2VPN и L3VPN? Варианты ответов:

    а) при оказании услуг L2VPN в сети провайдера связи используется технология второго уровня, а при оказании услуг L3VPN — третьего;

    б) при оказании услуг L2VPN провайдер соединяет сайты клиента на основе адресной информации второго уровня, а при оказании услуг L3VPN — третьего.

ГЛАВА 20 Технология MPLS

Технология **многопротокольной коммутации с помощью меток** (MultiProtocol Label Switching, MPLS) считается сегодня многими специалистами одной из самых перспективных транспортных технологий. Эта технология объединяет технику виртуальных каналов с функциональностью стека TCP/IP.

Объединение происходит за счет того, что одно и то же сетевое устройство, называемое **коммутирующим по меткам маршрутизатором** (Label Switch Router, LSR), выполняет функции как IP-маршрутизатора, так и коммутатора виртуальных каналов. Причем это не механическое объединение двух устройств, а тесная интеграция, когда функции каждого устройства дополняют друг друга и используются совместно.

Многопротокольность технологии MPLS состоит в том, что она позволяет использовать протоколы маршрутизации не только стека TCP/IP, но и любого другого стека, например IPX/SPX. В этом случае вместо протоколов маршрутизации RIP IP, OSPF и IS-IS применяется протокол RIP IPX или NLSP, а общая архитектура LSR останется такой же. Во времена разработки технологии MPLS в середине 90-х годов, когда на практике функционировало несколько стеков протоколов, такая многопротокольность представлялась важной, однако сегодня в условиях доминирования стека протоколов TCP/IP это свойство уже не является значимым. Правда, сегодня многопротокольность MPLS можно понимать по-другому — как свойство передавать с помощью соединений MPLS трафик разных протоколов канального уровня; это свойство MPLS рассматривается в главе 21.

Главное достоинство MPLS видится сегодня многими специалистами в способности предоставлять разнообразные транспортные услуги в IP-сетях, в первую очередь — услуги виртуальных частных сетей. Эти услуги отличаются разнообразием, они могут предоставляться как на сетевом, так и на канальном уровне. Кроме того, MPLS дополняет дейтаграммные IP-сети таким важным свойством, как передача трафика в соответствии с техникой виртуальных каналов, что позволяет выбирать нужный режим передачи трафика в зависимости от требований услуги. Виртуальные каналы MPLS обеспечивают инжиниринг трафика, так как они поддерживают детерминированные маршруты.

Базовые принципы и механизмы MPLS

Совмещение коммутации и маршрутизации в одном устройстве

Впервые идея объединения маршрутизации и коммутации в одном устройстве была реализована в середине 90-х годов компанией *Ipsilon*, которая начала выпускать комбинированные устройства IP/ATM. В этих устройствах была реализована новая технология **IP-коммутации** (IP switching), которая решала проблему неэффективной передачи кратковременных потоков данных в сетях ATM, которые в то время стали широко использоваться для передачи компьютерных данных в сетях операторов связи. ATM-коммутаторы существенно превосходили IP-маршрутизаторы по производительности, поэтому провайдеры при обработке IP-трафика старались применять как можно меньше промежуточных маршрутизаторов, передавая трафик между ними через быстрые ATM-коммутаторы.

Проблема передачи кратковременных потоков состоит в том, что для них нет смысла создавать постоянный виртуальный канал (PVC), так как поток данных между двумя конкретными абонентами существует лишь короткое время, и созданный виртуальный канал подавляющую часть времени используется провайдером не по назначению. Аналогом такой ситуации может быть телефонная сеть, в которой для каждого абонента создано постоянное соединение со всеми его возможными собеседниками. Казалось бы, технология ATM предлагает готовый ответ — именно для таких ситуаций и были предусмотрены коммутируемые виртуальные каналы (SVC). Однако в случае, когда время установления соединения SVC равно или даже превосходит время передачи данных, эффективность коммутируемых виртуальных каналов также оказывается невысокой. Это очень напоминает ситуацию, когда для того, чтобы поговорить 5 минут по телефону, требовалось бы всякий раз затрачивать 5 минут на дозвон до нужного абонента. А в ATM-коммутаторах часто наблюдалась именно такая ситуация, так как время пульсации компьютерного трафика было соизмеримо со временем установления соединения SVC.

В качестве решения проблемы компания Ipsilon предложила встроить во все ATM-коммутаторы блоки IP, которые поддерживали протокол IP для продвижения пакетов на основе IP-адресов, и протоколы маршрутизации стека TCP/IP для автоматического построения таблиц маршрутизации. В сущности, к ATM-коммутатору был добавлен IP-маршрутизатор.

Передача IP-пакета, принадлежащего кратковременному потоку, осуществлялась по сети Ipsilon следующим образом. Пакет поступал от узла-отправителя на комбинированное устройство IP/ATM, которое разбивало этот пакет на ATM-ячейки. Каждая ячейка кратковременного потока затем инкапсулировалась в новый IP-пакет, который передавался от одного устройства IP/ATM к другому, а затем к адресату по маршруту, определяемому обычными таблицами IP-маршрутизации, хранящимися в этих устройствах.

При этом стандартное для технологии ATM виртуальное соединение между устройствами IP/ATM не устанавливалось, а передача кратковременных IP-потоков существенно ускорялась за счет исключения времени установления соединения SVC. Долговременные потоки передавались устройствами IP/ATM традиционным для ATM способом — с помощью виртуальных каналов PVC или SVC. Так как топология сети является одной и той же как для протоколов IP, так и для протоколов ATM, появляется возможность использовать один и тот же протокол маршрутизации для обеих частей комбинированного устройства.

Для реализации своей технологии компания Ipsilon встроила в устройства IP/ATM фирменные протоколы, ответственные за распознавание длительности потоков данных и установление виртуальных каналов для долговременных потоков. Эти протоколы были оформлены в виде проектов стандартов Интернета, но стандартами Интернета не стали.

Технология IP-коммутации была разработана для сетей операторов связи. Эти сети принимают на границе с другими сетями IP-трафик и ускоренно передают его через свою магистраль. Важным обстоятельством здесь является то, что одни поставщики услуг Интернета (ISP) могут применять эту технологию *независимо* от других, оставаясь для внешнего мира операторами обычной IP-сети.

Технология IP-коммутации была сразу замечена операторами связи и стала достаточно популярной. Инициативу Ipsilon развила компании *Cisco Systems*, создав собственную **технологию коммутации на основе тегов** (tag switching), которая явилась значительным шагом вперед на пути объединения протоколов IP с техникой виртуальных соединений, однако она, так же как и IP-коммутация, не стала стандартной технологией.

На базе этих фирменных технологий рабочая группа IETF, состоящая из специалистов различных компаний, создала в конце 90-х годов технологию MPLS.

В MPLS был сохранен главный принцип технологий-предшественниц.

> В одном и том же устройстве поддерживается два разных способа продвижения пакетов: дейтаграммный на основе IP-адресов и ориентированный на соединения механизм виртуальных каналов. В то же время протоколы маршрутизации используются для определения топологии сети и автоматического построения таблиц IP-маршутизации и таблиц MPLS-продвижения. Комбинированное устройство может задействовать любой из двух способов продвижения пакетов в зависимости от конфигурационных параметров протокола MPLS.

Принцип объединения протоколов различных технологий иллюстрируют рис. 20.1 и 20.2. На первом из них показана упрощенная архитектура стандартного IP-маршрутизатора, на втором — архитектура комбинированного устройства LSR, поддерживающего технологии IP и MPLS.

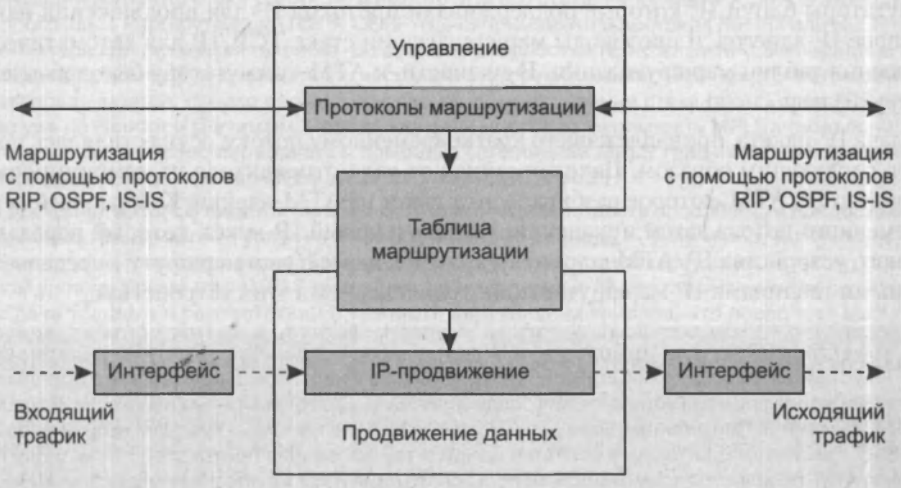


Рис. 20.1. Архитектура IP-маршрутизатора

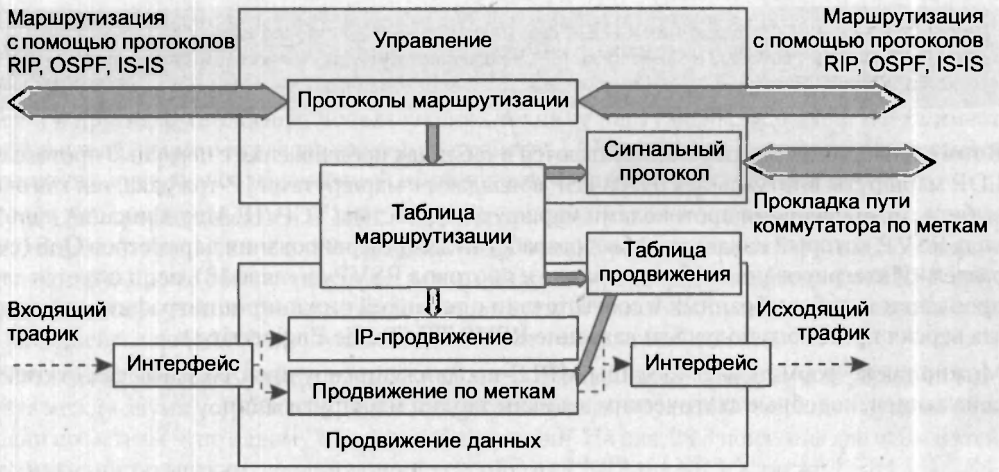


Рис. 20.2. Архитектура LSR

Так как устройство LSR выполняет все функции IP-маршрутизатора, оно содержит все блоки последнего, а для поддержки функций MPLS в LSR включен ряд дополнительных блоков, относящихся как к управлению, так и к продвижению данных.

В качестве примера можно указать на *блок продвижения по меткам*, который передает IP-пакет не на основе IP-адреса назначения, а на основе поля метки. При принятии решения о выборе следующего хопа блок продвижения по меткам использует *таблицу коммутации*, которая в стандарте MPLS носит название **таблицы продвижения**. Таблица продвижения в технологии MPLS похожа на аналогичные таблицы других технологий, основанных на технике виртуальных каналов (табл. 20.1).

Таблица 20.1. Пример таблицы продвижения в технологии MPLS

Входной интерфейс	Метка	Следующий хоп	Действия
S0	245	S1	256
S0	27	S2	45
...	...	...	...

Внимательный читатель заметил, наверное, небольшое отличие данной таблицы от таблицы коммутации Frame Realy, представленной на рис. 19.8. Действительно, вместо поля выходного интерфейса здесь поле следующего хопа, а вместо поля выходной метки — поле действий. В большинстве случаев обработки MPLS-кадров эти поля используются точно таким же образом, как соответствующие им поля обобщенной таблицы коммутации. То есть значение поля следующего хопа является значением интерфейса, на который нужно передать кадр, а значение поля действий — новым значением метки. Однако в некоторых случаях эти поля служат другим целям, о чем будет сказано позже.

Рассматриваемые таблицы для каждого устройства LSR формируются *сигнальным протоколом*. В MPLS используется два различных сигнальных протокола: **протокол распределения меток** (Label Distribution Protocol, LDP) и модификация уже знакомого нам протокола резервирования ресурсов RSVP.

Формируя таблицы продвижения на LSR, сигнальный протокол прокладывает через сеть виртуальные маршруты, которые в технологии MPLS называют **путями коммутации по меткам** (Label Switching Path, LSP).

В том случае, когда метки устанавливаются в таблицах продвижения с помощью протокола LDP, маршруты виртуальных путей LSP совпадают с маршрутами IP-трафика, так как они выбираются обычными протоколами маршрутизации стека TCP/IP. Модификация протокола RSVP, который изначально был разработан для резервирования параметров QoS (см. раздел «Интегрированное обслуживание и протокол RSVP» в главе 18), используется для прокладки путей, выбранных в соответствии с техникой инжиниринга трафика, поэтому эта версия протокола получили название RSVP TE (Traffic Engineering).

Можно также формировать таблицы MPLS-продвижения вручную, создавая там статические записи, подобные статическим записям таблиц маршрутизации.

Пути коммутации по меткам

Архитектура MPLS-сети описана в RFC 3031 (http://www.rfc-editor.org/rfc/rfc3031.txt). Основные элементы этой архитектуры представлены на рис. 20.3, где MPLS-сеть взаимодействует с несколькими IP-сетями, возможно, не поддерживающими технологию MPLS.

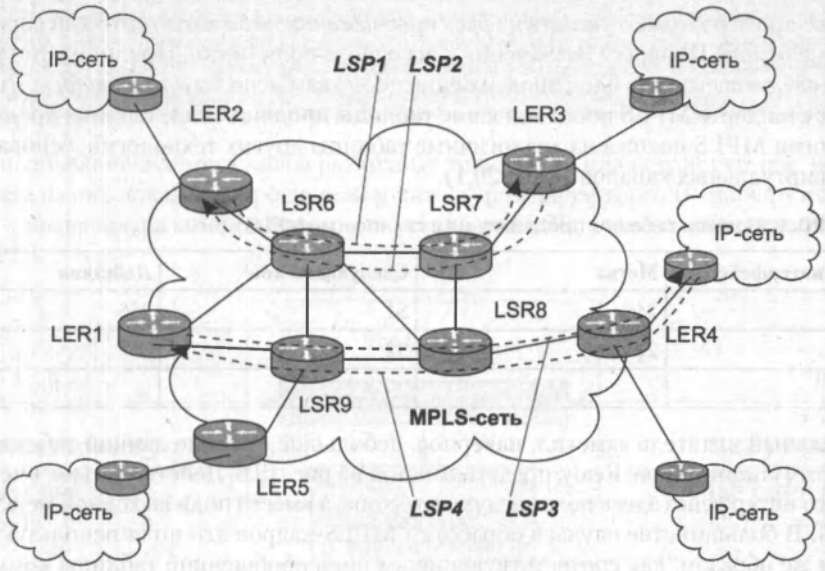


Рис. 20.3. MPLS-сеть

Пограничные устройства LSR в технологии MPLS имеют специальное название — **пограничные коммутирующие по меткам маршрутизаторы** (Label switch Edge Router, LER).

Устройство LER, являясь функционально более сложным, принимает трафик от других сетей в форме стандартных IP-пакетов, а затем добавляет к нему метку и направляет вдоль

соответствующего пути к выходному устройству LER через несколько промежуточных устройств LSR. При этом пакет продвигается не на основе IP-адреса назначения, а на основе метки.

Как и в других технологиях, использующих технику виртуальных каналов, метка имеет локальное значение в пределах каждого устройства LER и LSR, то есть при передаче пакета с входного интерфейса на выходной выполняется смена значения метки.

Пути LSP прокладываются в MPLS *предварительно* в соответствии с топологией сети, аналогично маршрутам для IP-трафика (и на основе работы тех же протоколов маршрутизации). Кроме того, существует режим инжиниринга трафика, когда пути LSP прокладываются с учетом требований к резервируемой для пути пропускной способности и имеющейся свободной пропускной способности каналов связи сети.

LSP представляет собой *однонаправленный* виртуальный канал, поэтому для передачи трафика между двумя устройствами LER нужно установить, по крайней мере, два пути коммутации по меткам — по одному в каждом направлении. На рис. 20.3 показаны две пары путей коммутации по меткам, соединяющие устройства LER2 и LER3, а также LER1 и LER4.

LER выполняет такую важную функцию, как направление входного трафика в один из исходящих из LER путей LSP. Для реализации этой функции в MPLS введено такое понятие, как **класс эквивалентности продвижения** (Forwarding Equivalence Class, FEC).

> Класс эквивалентности продвижения — это группа IP-пакетов, имеющих одни и те же требования к условиям транспортировки (транспортному сервису). Все пакеты, принадлежащие к данному классу, продвигаются через MPLS-сеть по одному виртуальному пути LSP.

В LER существует база данных классов FEC; каждый класс описывается набором элементов, а каждый элемент описывает признаки, на основании которых входящий пакет относят к тому или иному классу.

Классификация FEC может выполняться различными способами. Вот несколько примеров:

- *На основании IP-адреса назначения.* Это наиболее близкий к принципам работы IP-сетей подход, который состоит в том, что для каждого префикса сети назначения, имеющегося в таблице LER-маршрутизации, создается отдельный класс FEC. Протокол LDP, который мы далее рассмотрим, полностью автоматизирует процесс создания классов FEC по этому способу.
- *В соответствии с требованиями инжиниринга трафика.* Классы выбираются таким образом, чтобы добиться баланса загрузки каналов сети.
- *В соответствии с требованиями VPN.* Для конкретной виртуальной частной сети клиента создается отдельный класс FEC.
- *По типам приложений.* Например, трафик IP-телефонии (RTP) составляет один класс FEC, а веб-трафик — другой.
- *По интерфейсу, с которого получен пакет.*
- *По MAC-адресу назначения кадра*, если это кадр Ethernet.

Как видно из приведенных примеров, при классификации трафика в MPLS могут использоваться признаки не только из заголовка IP-пакета, но и многие другие, включая информацию канального (MAC-адрес) и физического (интерфейс) уровней.

После принятия решения о принадлежности пакета к определенному классу FEC его нужно связать с существующим путем LSP. Для этой операции LER использует таблицу FTN (FEC To Next hop — отображение класса FEC на следующий хоп). Таблица 20.2 представляет собой пример FTN.

Таблица 20.2. Пример FTN

Признаки FEC	Метка
123.20.0.0/16; 195.14.0.0/16	106
194.20.0.0/24; eth1	107

На основании таблицы FTN каждому входящему пакету назначается соответствующая метка, после чего этот пакет становится неразличим в домене MPLS от других пакетов того же класса FEC, все они продвигаются по одному и тому же пути внутри домена.

Сложная настройка и конфигурирование выполняются только в LER, а все промежуточные устройства LSR выполняют простую работу, продвигая пакет в соответствии с техникой виртуального канала.

Выходное устройство LER удаляет метку и передает пакет в следующую сеть уже в стандартной форме IP-пакета. Таким образом, технология MPLS остается прозрачной для остальных IP-сетей.

Обычно в MPLS-сетях используется усовершенствованный по сравнению с описанным алгоритм обработки пакетов. Усовершенствование заключается в том, что удаление метки выполняет не последнее на пути устройство, а *предпоследнее*. Действительно, после того как предпоследнее устройство определит на основе значения метки следующий хоп, метка в MPLS-кадре уже не нужна, так как последнее устройство, то есть выходное устройство LER, будет продвигать пакет на основе значения IP-адреса. Это небольшое изменение алгоритма продвижения кадра позволяет сэкономить одну операцию над MPLS-кадром. В противном случае последнее вдоль пути устройство должно было бы удалить метку, а уже затем выполнить просмотр таблицы IP-маршрутизации. Эта техника получила название техники **удаления метки на предпоследнем хопе** (Penultimate Hop Popping, PHP).

Заголовок MPLS и технологии канального уровня

Заголовок MPLS состоит из нескольких полей (рис. 20.4):

- *Метка* (20 бит). Используется для выбора соответствующего пути коммутации по меткам.
- *Время жизни* (TTL). Это поле, занимающее 8 бит, дублирует аналогичное поле IP-пакета. Это необходимо для того, чтобы устройства LSR могли отбрасывать «заблудившиеся» пакеты только на основании информации, содержащейся в заголовке MPLS, не обращаясь к заголовку IP.
- *Класс услуги* (Class of Service, CoS). Поле CoS, занимающее 3 бита, первоначально было зарезервировано для развития технологии, но в последнее время используется в основном для указания класса трафика, требующего определенного уровня QoS.
- *Признак дна стека меток*. Этот признак (S) занимает 1 бит.

Концепцию стека меток мы рассмотрим в следующем разделе, а пока для пояснения механизма взаимодействия MPLS с технологиями канального уровня рассмотрим ситуацию, когда заголовок MPLS включает только одну метку.

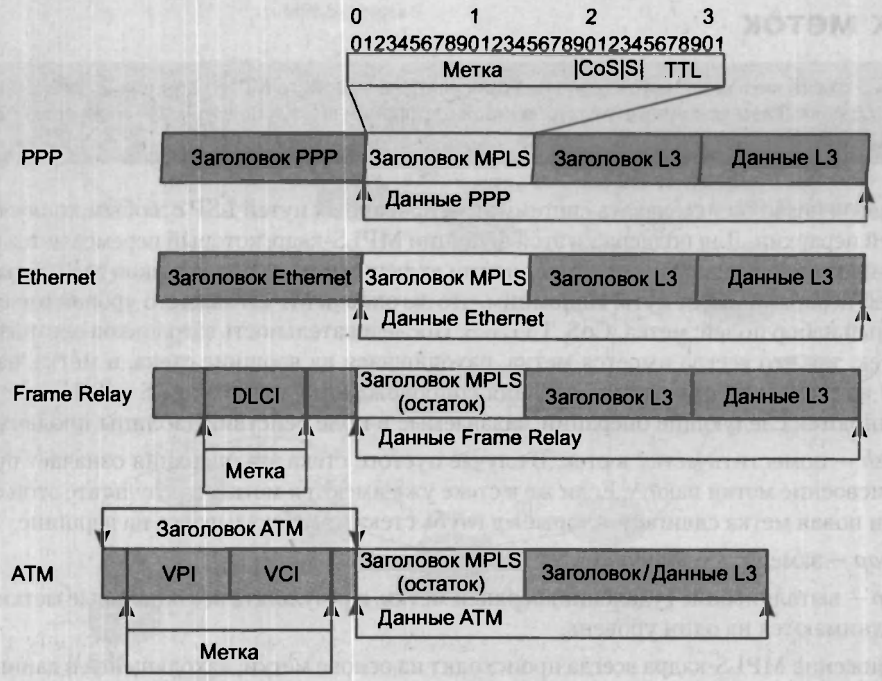


Рис. 20.4. Форматы заголовков нескольких разновидностей технологии MPLS

Как видно из рисунка, технология MPLS поддерживает несколько типов кадров: PPP, Ethernet, Frame Relay и ATM. Это не означает, что под слоем MPLS работает какая-либо из перечисленных технологий, например Ethernet. Это означает только то, что в технологии MPLS используются форматы кадров этих технологий для помещения в них пакета сетевого уровня, которым сегодня почти всегда является IP-пакет.

В связи с тем, что заголовок MPLS помещается между заголовком канального уровня и заголовком IP, его называют **заголовком-вставкой** (shim header).

Продвижение кадра в MPLS-сети происходит на основе метки MPLS и техники LSP, а не на основе адресной информации и техники той технологии, формат кадра которой MPLS использует. Таким образом, если в MPLS применяется кадр Ethernet, то MAC-адреса источника и приемника хотя и присутствуют в соответствующих полях кадра Ethernet, но для продвижения кадров не используются. Исключение составляет случай, когда между двумя соседними устройствами LSR находится сеть коммутаторов Ethernet — тогда MAC-адрес назначения MPLS-кадра потребуется для того, чтобы кадр дошел до следующего устройства LSR, а уже оно будет продвигать его на основании метки.

В кадрах PPP, Ethernet и Frame Relay заголовок MPLS помещается между оригинальным заголовком и заголовком пакета 3-го уровня. С ячейками ATM технология MPLS поступает по-другому: она пользуется имеющимися полями VPI/VCI в заголовках этих ячеек для меток виртуальных соединений. Поля VPI/VCI нужны только для хранения поля метки, остальная часть заголовка MPLS с полями CoS, S и TTL размещается в поле данных ATM-ячеек и при передаче ячеек ATM-коммутаторами, поддерживающими технологию MPLS, не используется.

Далее для определенности при рассмотрении примеров мы будем подразумевать, что используется формат кадров MPLS/PPP.

Стек меток

Наличие **стека меток** является одним из оригинальных свойств MPLS. Концепция стека меток является развитием концепции двухуровневой адресации виртуальных путей с помощью меток VPI/VCI, принятой в ATM.

Стек меток позволяет создавать систему агрегированных путей LSP с любым количеством уровней иерархии. Для поддержки этой функции MPLS-кадр, который перемещается вдоль иерархически организованного пути, должен включать столько заголовков MPLS, сколько уровней иерархии имеет путь. Напомним, что заголовок MPLS каждого уровня имеет собственный набор полей: метка, CoS, TTL и S. Последовательность заголовков организована как стек, так что всегда имеется метка, находящаяся на вершине стека, и метка, находящаяся на дне стека, при этом последняя сопровождается признаком S = 1. Над метками выполняются следующие операции, задаваемые в поле действий таблицы продвижения:

- *Push* — поместить метку в стек. В случае пустого стека эта операция означает простое присвоение метки пакету. Если же в стеке уже имеются метки, в результате этой операции новая метка сдвигает «старые» в глубь стека, сама оказываясь на вершине.
- *Swap* — заменить текущую метку новой.
- *Pop* — выталкивание (удаление) верхней метки, в результате все остальные метки стека поднимаются на один уровень.

Продвижение MPLS-кадра всегда происходит на основе метки, находящейся в данный момент на вершине стека. Рассмотрим сначала продвижение MPLS-кадра по *одноуровневому* пути в MPLS-сети, показанной на рис. 20.5.

Сеть состоит из трех MPLS-доменов. На рисунке показаны путь LSP1 в домене 1 и путь LSP2 в домене 2. LSP1 соединяет устройства LER1 и LER2, проходя через устройства LSR1, LSR2 и LSR3. Пусть начальной меткой пути LSP1 является метка 256, которая была присвоена пакету пограничным устройством LER1. На основании этой метки пакет поступает на устройство LSR1, которое по своей таблице продвижения определяет новое значение метки пакета (272) и переправляет его на вход LSR2. Устройство LSR2, действуя аналогично, присваивает пакету новое значение метки (132) и передает его на вход LSR3. Устройство LSR3, будучи предпоследним устройством в пути LSP1, выполняет операцию *Pop* и удаляет метку из стека. Устройство LER2 продвигает пакет уже на основании IP-адреса.

На рисунке также показан путь LSP2 в домене 2. Он соединяет устройства LER3 и LER4, проходя через устройства LSR4, LSR5 и LSR6, и определяется последовательностью меток 188, 112, 101.

Для того чтобы IP-пакеты могли передаваться на основе техники MPLS не только внутри каждого домена, но и между доменами (например, между устройствами LER1 и LER4), существует два принципиально разных решения.

- Первое решение состоит в том, что между LER1 и LER4 устанавливается один *одноуровневый* путь коммутации по меткам, соединяющий пути LSP1 и LSP2 (которые в этом случае становятся одним путем). Это простое, на первый взгляд, решение, называемое **сшиванием** путей LSP, плохо работает в том случае, когда MPLS-домены принадлежат разным поставщикам услуг, не позволяя им действовать независимо друг от друга.
- Вторым более перспективным решением является применение *многоуровневого* подхода к соединению двух MPLS-доменов, принадлежащих, возможно, разным поставщикам услуг.

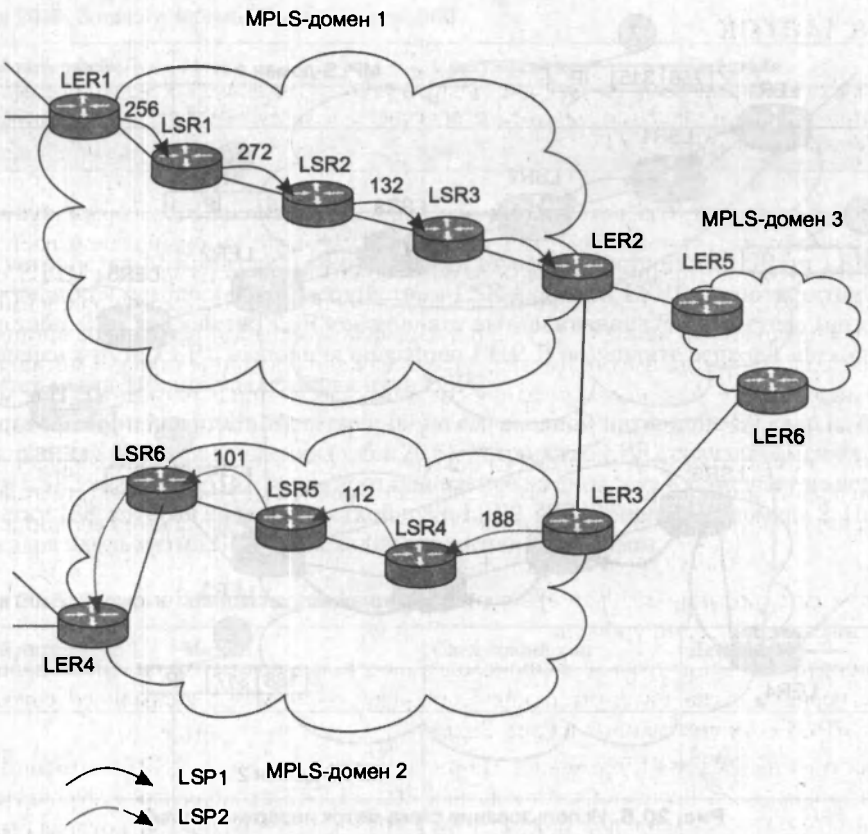


Рис. 20.5. Пути LSP1 и LSP2, проложенные в доменах 1 и 2 MPLS-сети

Для реализации второго подхода в нашем примере нужно создать путь коммутации по меткам второго уровня (LSP3), соединяющий устройства LER1 и LER4. Этот путь определяет последовательность хопов *между доменами*, а не между внутренними устройствами LSR каждого домена. Так, LSP3 состоит из хопов LER1 – LER2 – LER3 – LSR4. В этом отношении многоуровневый подход MPLS концептуально очень близок подходу протокола BGP, определяющего путь между автономными системами.

Рассмотри более детально, как работает технология MPLS в случае путей коммутации по меткам двух уровней (рис. 20.6).

В устройстве LER1 начинаются два пути – LSP1 и LSP3 (последний показан на рисунке серым цветом), что обеспечивается соответствующей записью в таблице продвижения устройства LER1 (табл. 20.3).

Таблица 20.3. Запись в таблице продвижения LER1

Входной интерфейс	Метка	Следующий хоп	Действия
...	...	...	...
S0	–	S1	315 Push 256
...	...	...	...

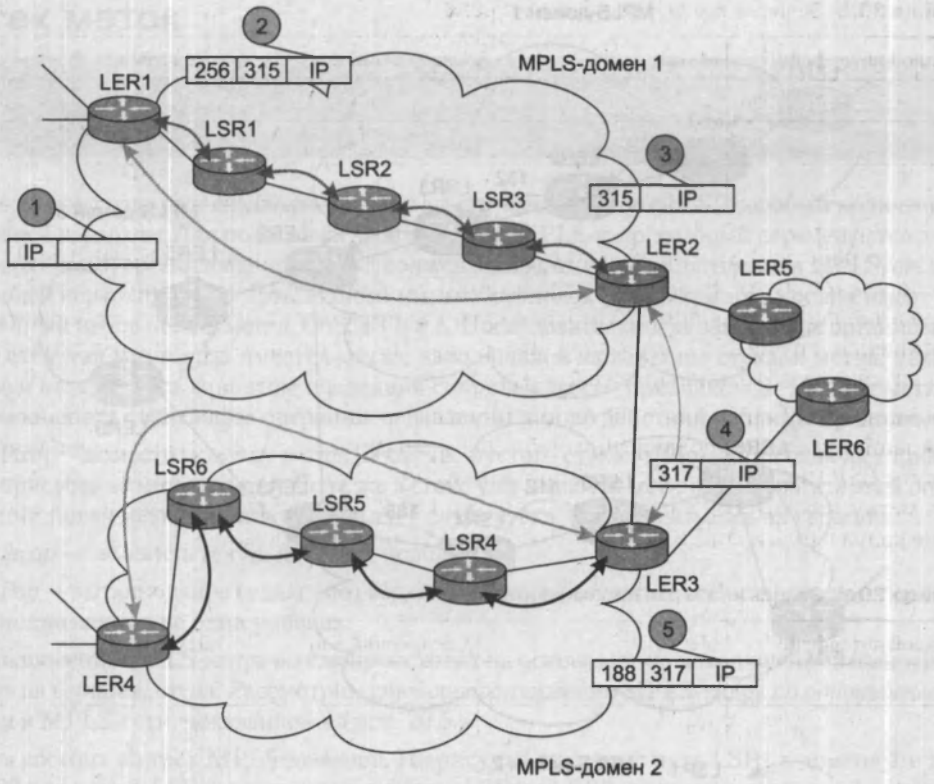


Рис. 20.6. Использование стека меток иерархией путей

IP-пакеты, поступающие на интерфейс S0 устройства LER1, продвигаются на его выходной интерфейс S1, где для них создается заголовок MPLS, включающий метку 315 верхнего уровня (LSP3), которая на этот момент является верхушкой стека меток. Затем эта метка проталкивается на дно стека (операция *Push*), а верхней становится метка 256, относящаяся к LSP1.

Далее MPLS-кадр с меткой 256 поступает на выходной интерфейс S1 пограничного устройства LER1 и передается на вход LSR1. Устройство LSR1 обрабатывает кадр в соответствии со своей таблицей продвижения (табл. 20.4). Метка 256, находящаяся на вершине стека, заменяется меткой 272. (Отметьте, что метка 315, находящаяся ниже в стеке, устройством LSR1 игнорируется.)

Таблица 20.4. Запись в таблице продвижения LSR1

Входной интерфейс	Метка	Следующий хоп	Действия
...	...	...	...
S0	256	S1	272
...	...	...	...

Аналогичные действия выполняет устройство LSR2, которое заменяет метку меткой 132 и отправляет кадр следующему по пути устройству LSR3 (табл. 20.5).

Таблица 20.5. Запись в таблице продвижения LSR3

Входной интерфейс	Метка	Следующий хоп	Действия
...	...	...	...
S0	132	S1	Pop
...	...	...	...

Работа устройства LSR3 несколько отличается от работы устройств LSR1 и LSR2, так как оно является *предпоследним* устройством LSR для пути LSP1. В соответствии с записью в табл. 22.4 устройство LSR3 выполняет выталкивание (*Pop*) из стека метки 132, относящейся к пути LSP1, выполняя операцию PHP. В результате верхней меткой стека становится метка 315, принадлежащая пути LSP3.

Устройство LER2 продвигает поступивший на его входной интерфейс S0 кадр на основе своей записи таблицы продвижения (табл. 20.6). Устройство LER2 сначала заменяет метку 315 пути LSP3 значением 317, затем проталкивает ее на дно стека и помещает на вершину стека метку 188, которая является меткой пути LSP2, внутреннего для домена 2. Перемещение кадра вдоль пути LSP2 происходит аналогичным образом.

Таблица 20.6. Запись в таблице продвижения LER2

Входной интерфейс	Метка	Следующий хоп	Действия
...	...	...	...
S0	315	S1	317 Push 188
...	...	...	...

Описанная модель двухуровневого пути легко может быть расширена для любого количества уровней.

Протокол LDP

Протокол распределения меток (Label Distribution Protocol, LDP) позволяет автоматически создавать в сети пути LSP в соответствии с существующими в таблицах маршрутизации записях о маршрутах в IP-сети. Протокол LDP принимает во внимание только те записи таблицы маршрутизации, которые созданы с помощью внутренних протоколов маршрутизации, то есть протоколов типа IGP, поэтому режим автоматического создания LSP с помощью протокола LDP иногда называют режимом MPLS IGP (в отличие от режима MPLS TE, когда маршруты выбираются из соображений инжиниринга трафика и не совпадают с маршрутами, выбранными внутренними протоколами маршрутизации). Еще режим MPLS IGP называют **ускоренной MPLS-коммутацией**, это название отражает начальную цель разработчиков технологии MPLS, которая состояла только в ускорении продвижения IP-пакетов с помощью техники виртуальных каналов. Спецификация LDP дается в RFC 5036 (http://www.rfc-editor.org/rfc/rfc5036.txt).

Рассмотрим работу протокола LDP на примере сети, изображенной на рис. 20.7.

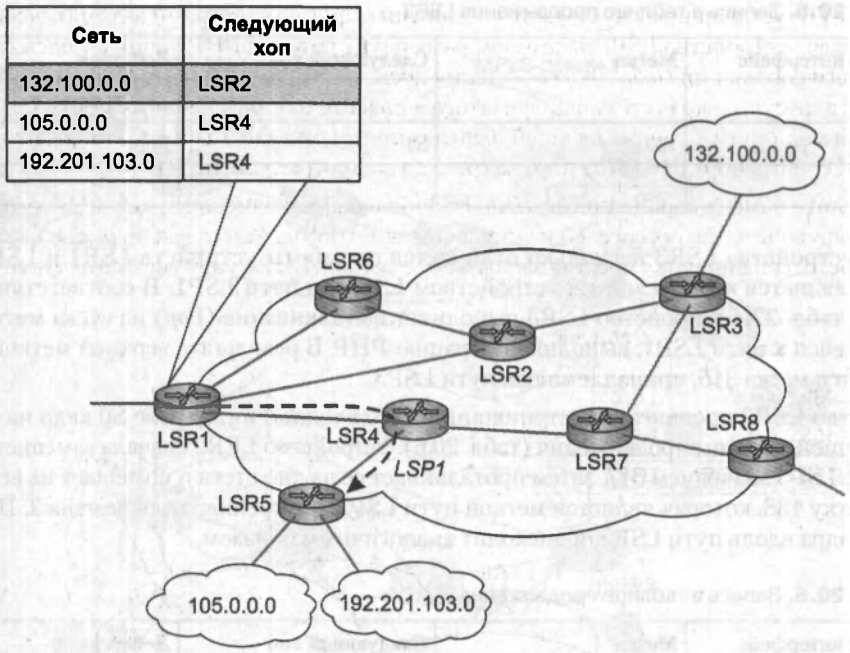


Рис. 20.7. MPLS-сеть с устройствами LSR, поддерживающими LDP

Все устройства LSR поддерживают сигнальный *протокол распределения меток* (LDP). От устройства LSR1 в сети уже установлен один путь LSP1 — по этому пути идет трафик к сетям 105.0.0.0 и 192.201.103.0. Это значит, что таблица FTN (отображающая сети назначения на LSP) у LSR1 соответствует табл. 20.7.

Таблица 20.7. Таблица FTN устройства LSR1

Признаки FEC	Метка
105.0.0.0; 192.201.103.0	231

Метка 231 в этой таблице соответствует пути LSP1.

Мы рассмотрим функционирование протокола LDP в ситуации, когда в результате работы протоколов маршрутизации или же после ручной модификации администратором сети в таблице маршрутизации устройства LSR1 появилась запись о новой сети назначения, для которой в сети поставщика услуг еще не проложен путь коммутации по меткам. В нашем случае это сеть 132.100.0.0 и для нее нет записи в таблице FTN.

В этом случае устройство LSR1 автоматически инициирует процедуру прокладки нового пути. Для этого оно запрашивает по протоколу LDP метку для новой сети 132.100.0.0 у маршрутизатора, IP-адрес которого в таблице маршрутизации указан для данной сети как адрес следующего хопа.

Однако для того чтобы воспользоваться протоколом LDP, нужно сначала установить между устройствами LSR сеанс LDP, так как этот протокол работает в режиме установления соединений.

Сеансы LDP устанавливаются между соседними маршрутизаторами автоматически. Для этого каждое устройство LSR, на котором развернут протокол LDP, начинает посылать своим соседям сообщения *Hello*. Эти сообщения посылают по групповому IP-адресу 224.0.0.2, который адресуется ко всем маршрутизаторам подсети и определенному порту UDP. Если соседний маршрутизатор также поддерживает протокол LDP, то он в ответ устанавливает сеанс TCP через порт 646 (этот порт закреплен за протоколом LDP).

В результате обмена сообщениями *Hello* все поддерживающие протокол LDP устройства LSR обнаруживают своих соседей и устанавливают с ними сеансы, как показано на рис. 20.8 (для простоты на рисунке представлены не все сеансы LDP, существующие в сети).

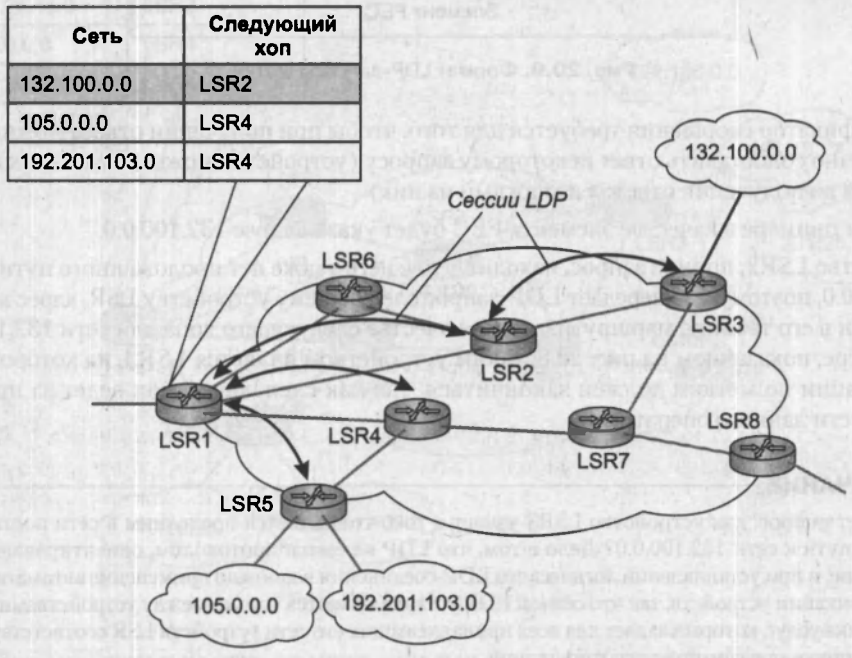


Рис. 20.8. Сеансы LDP устанавливаются между непосредственными соседями

Будем считать, что между устройствами LSR1 и LSR2 установлен сеанс LDP.

Тогда при обнаружении новой записи в таблице маршрутизации, указывающей на устройство LSR2 в качестве следующего хопа, устройство LSR1 просит устройство LSR2 назначить метку для нового пути к сети 132.100.0.0. Говорят, что устройство LSR2 находится ниже по потоку (downstream) для устройства LSR1 относительно пути к сети 132.100.0.0. Соответственно устройство LSR1 расположено выше по потоку для устройства LSR2 относительно сети 132.100.0.0. Естественно, что для других сетей назначения у устройства LSR1 имеются другие соседи вниз по потоку, а у устройства LSR2 – другие соседи вверх по потоку.

Причина, по которой значение метки для нового пути выбирается соседом ниже по потоку, понятна — эта метка, которая имеет локальное значение на двухточечном соединении между соседними устройствами, будет использоваться именно этим устройством для того, чтобы понимать, к какому пути LSP относится пришедший MPLS-кадр. Поэтому

устройство ниже по потоку выбирает уникальное значение метки, исходя из неиспользованных значений меток для своего интерфейса, который связывает его с соседом выше по потоку.

Для получения значения метки устройство LSR1 выполняет запрос метки протокола LDP. Формат такого запроса достаточно прост (рис. 20.9).

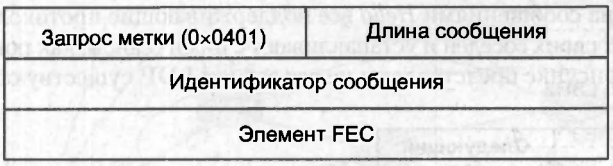


Рис. 20.9. Формат LDP-запроса метки

Идентификатор сообщения требуется для того, чтобы при получении ответа можно было однозначно сопоставить ответ некоторому запросу (устройство может послать несколько запросов до получения ответов на каждый из них).

В нашем примере в качестве элемента FEC будет указан адрес 132.100.0.0.

Устройство LSR2, приняв запрос, находит, что у него также нет проложенного пути к сети 132.100.0.0, поэтому оно передает LDP-запрос следующему устройству LSR, адрес которого указан в его таблице маршрутизации в качестве следующего хопа для сети 132.100.0.0. В примере, показанном на рис. 20.8, таким устройством является LSR3, на котором путь коммутации по меткам должен закончиться, так как следующий хоп ведет за пределы MPLS-сети данного оператора.

ПРИМЕЧАНИЕ

Возникает вопрос, как устройство LSR3 узнает о том, что является последним в сети поставщика услуг на пути к сети 132.100.0.0? Дело в том, что LDP является протоколом, ориентированным на соединение, и при установлении логического LDP-соединения возможно применение автоматической аутентификации устройств, так что сеансы LDP устанавливаются только между устройствами одного поставщика услуг, который задает для всех принадлежащих его сети устройств LSR соответствующую информацию для взаимной аутентификации.

Устройство LSR3, обнаружив, что для пути к сети 132.100.0.0 оно является пограничным, назначает для прокладываемого пути метку, еще не занятую его входным интерфейсом S0, и сообщает об этой метке устройству LSR2 в LDP-сообщении, формат которого представлен на рис. 20.10. Пусть это будет метка 231.

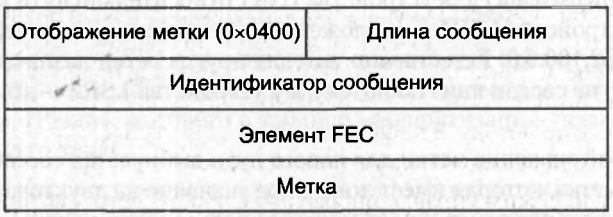


Рис. 20.10. Формат отображения метки на элемент FEC протокола LDP

В свою очередь, LSR2 назначает неиспользуемую его интерфейсом S0 метку и сообщает об этом в LDP-сообщении отображения метки устройству LSR1. После этого новый путь коммутации по меткам, ведущий от LSR1 к сети 132.100.0.0, считается проложенным (рис. 20.11), и вдоль него пакеты начинают передаваться уже на основе меток и таблиц продвижения, а не IP-адресов и таблиц маршрутизации.

Сеть	Следующий хоп
132.100.0.0	LSR2
194.15.17.0	LSR2
201.25.10.0	LSR2
105.0.0.0	LSR4
192.201.103.0	LSR4

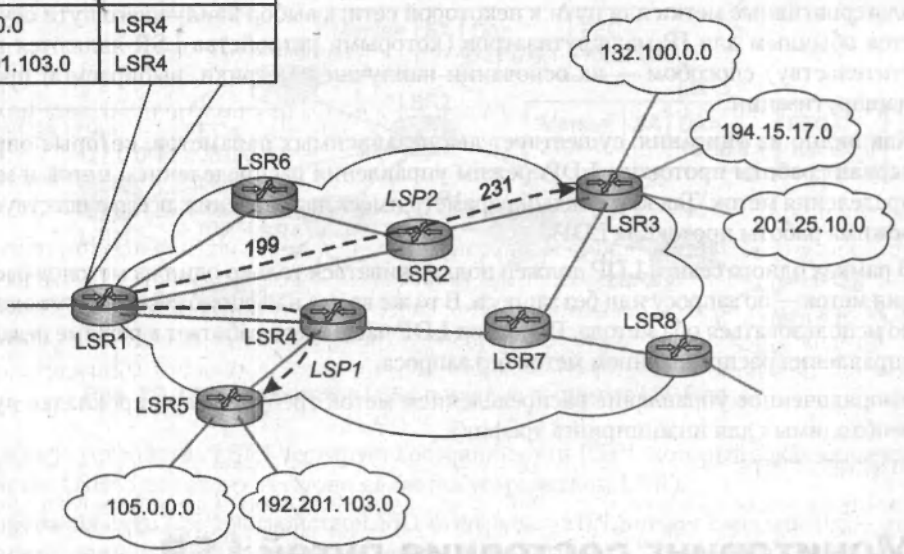


Рис. 20.11. Новый путь LSP2

Было бы нерационально прокладывать отдельный путь для каждой сети назначения каждого маршрутизатора. Поэтому устройства LSR стараются строить агрегированные пути коммутации по меткам и передавать вдоль них пакеты, следующие к некоторому набору сетей. Так, на рис. 20.11 устройство LSR1 передает по пути LSP1 пакеты, следующие не только к сети 132.100.0.0, но и к сетям 194.15.17.0 и 201.25.10.0, информация о которых появилась уже после того, как путь LSP2 был проложен.

Мы рассмотрели только один режим работы протокола LDP, который носит сложное название «Упорядоченный режим управления распределением меток с запросом устройства вниз по потоку». Здесь под упорядоченным режимом понимается такой режим, когда некоторое промежуточное устройство LSR не передает метку для нового пути устройству LSR, лежащему выше по потоку, до тех пор, пока не получит метку для этого пути от устройства LSR, лежащего ниже по потоку. В нашем случае устройство LSR2 ждало получения метки от LSR3 и уже потом передало метку устройству LSR1.

Существует и другой режим управления распределением меток, который называется независимым. При независимом управлении распределением меток LSR может назначить

и передать метку, не дожидаясь прихода сообщения от своего соседа, лежащего ниже по потоку. Например, устройство LSR2 могло бы назначить и передать метку 199 устройству LSR1, не дожидаясь прихода метки 231 от устройства LSR3. Так как метки имеют локальное значение, результат изменения режима не изменился бы.

Существует также два метода распределения меток — распределение от лежащего ниже по потоку по запросу и без запроса. Для нашего случая это значит, что если бы устройство LSR2 обнаружило в своей таблице маршрутизации запись о новой сети 132.100.0.0, оно могло бы назначить метку новому пути и передать ее устройству LSR1 без запроса. Так как при этом устройство LSR2 не знает своего соседа выше по потоку (таблица маршрутизации не говорит об этом), оно передает эту информацию всем своим соседям по сеансам LDP. В этом варианте работы протокола LDP устройства LSR могут получать альтернативные метки для пути к некоторой сети; а выбор наилучшего пути осуществляется обычным для IP-маршрутизатров (которыми устройства LSR являются по совместительству) способом — на основании наилучшей метрики, выбираемой протоколом маршрутизации.

Как видно из описания, существует два независимых параметра, которые определяют вариант работы протокола LDP: режим управления распределением меток и метод распределения меток. Так как каждый параметр имеет два значения, всего существует четыре режима работы протокола LDP.

В рамках одного сеанса LDP должен поддерживаться только один из методов распределения меток — по запросу или без запроса. В то же время в масштабах сети могут одновременно использоваться оба метода. Протокол LDP чаще всего работает в режиме независимого управления распределением меток без запроса.

Упорядоченное управление распределением меток требуется при прокладке путей LSP, необходимых для инжиниринга трафика.

Мониторинг состояния путей LSP

Наличие встроенных в транспортную технологию средств мониторинга состояния соединений и локализации ошибок (то есть средств OAM) является необходимым условием для того, чтобы она претендовала на статус технологии операторского класса. В противном случае ее трудно будет использовать операторам сетей, которым нужно обеспечивать своих многочисленных клиентов транспортным сервисом с высоким коэффициентом готовности (в пределах 0,999–0,99999), как это принято в телекоммуникационных сетях.

Первоначально технология MPLS не имела таких встроенных средств, полагаясь на такие средства стека TCP/IP, как утилиты ping и traceroute (использующие, как вы знаете из главы 17, ICMP-сообщения *Echo Request* и *Echo Response*). Однако классические утилиты ping и traceroute стека TCP/IP не дают корректной информации о состоянии путей LSP, так как они могут переноситься как вдоль, так и в обход этих путей с помощью обычной техники продвижения пакетов протокола IP. Поэтому позднее был разработан специальный протокол LSP Ping, который позволяет как тестировать работоспособность LSP (режим *ping*), так и локализовывать отказы (режим *traceroute*).

Кроме того, для мониторинга состояния LSP можно применять более экономичный, чем LSP Ping, протокол двунаправленного обнаружения ошибок продвижения (см. далее).

Тестирование путей LSP

В протоколе LSP Ping для тестирования состояния LSP применяется техника, близкая к механизму работы утилиты ping протокола IP. Она заключается в том, что протокол LSP Ping отправляет вдоль тестируемого пути LSP сообщение *Echo Request*. Если такое сообщение доходит до устройства LER, которое является конечным узлом тестируемого пути LSP, оно отвечает сообщением *Echo Replay*. Получение исходным узлом такого сообщения означает, что путь LSP работоспособен.

Описанная схема работы аналогична схеме работы утилиты ping протокола IP, однако она имеет свои особенности, которые мы поясним на примере сети, изображенной на рис. 20.12.

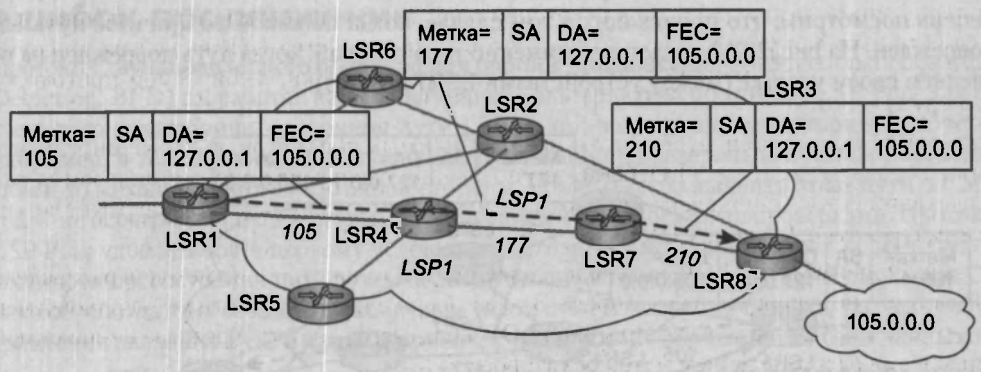


Рис. 20.12. Тестирование LSP с помощью протокола LSP Ping

В этом примере устройство LSR1 тестирует состояние пути LSP1, который заканчивается на устройстве LSR8 (для этого пути оно является устройством LER).

Для тестирования пути LSP1 устройство LSR1 отправляет MPLS-пакет с меткой 105 — эта метка соответствует пути LSP1 на линии между устройствами LSR1 и LSR4. Сообщение *Echo Request* вкладывается в UDP-сообщение, которое, в свою очередь, вкладывается в IP-пакет. На рис. 20.12 показаны только значимые для изучения протокола LSP Ping поля: метка MPLS-кадра, IP-адрес источника (SA), IP-адрес назначения (DA), а также поле FEC, которое идентифицирует тестируемый путь LSP. В нашем примере это IP-адрес сети 105.0.0.0, к которой ведет путь LSP1.

Адрес назначения в IP-пакете, который переносит сообщение *Echo Request*, равен 127.0.0.1, то есть является адресом обратной петли стека протоколов IP каждого узла. О причине использования такого необычного адреса назначения (а не, скажем, IP-адреса интерфейса конечного узла тестируемого пути LSP) мы расскажем позже, а пока заметим, что адрес 127.0.0.1 должен работать правильно, так как в процессе передачи запроса по сети для его продвижения используются MPLS-метки, а не IP-адрес назначения. При приходе на конечный узел IP-пакет освобождается от заголовка MPLS (это также может произойти на предыдущем хопе, если применяется техника PHP) и обрабатывается на основе IP-адреса. Так как адрес 127.0.0.1 указывает на собственный узел, то пакет передается собственному стеку TCP/IP, где он распознается как UDP-пакет протокола LSP Ping и обрабатывается соответственно.

Поле FEC посылается в запросе *Echo Request* для того, чтобы конечный узел пути мог сравнить указанное в пакете значение FEC со значением из его собственной базы данных

для пути, по которому пришел кадр запроса. Такой механизм позволяет отслеживать ситуации, когда запрос вследствие каких-то ошибок приходит не по тому пути, который тестируется.

В том случае, когда запрос благополучно доходит до конечного узла пути, и тот убеждается, что полученный запрос пришел по нужному пути (то есть полученное значение FEC совпадает со значением FEC из базы данных конечного узла), он отправляет ответ *Echo Replay* узлу, выполнившему запрос. В нашем случае узел LSR8 отправляет ответ *Echo Replay* узлу LSR1. Сообщение *Echo Replay* посылается уже не по пути LSP, а как обычное UDP-сообщение, вложенное в IP-пакет. Если вспомнить, что пути LSP являются однонаправленными, станет понятно, что это единственное гарантированное решение, так как обратного пути от LSR8 к LSR1 может и не существовать.

Теперь посмотрим, что происходит в том случае, когда по какой-то причине путь LSP поврежден. На рис. 20.13 представлен именно такой случай, когда путь поврежден на последнем своем участке (между устройствами LSR7 и LSR8).

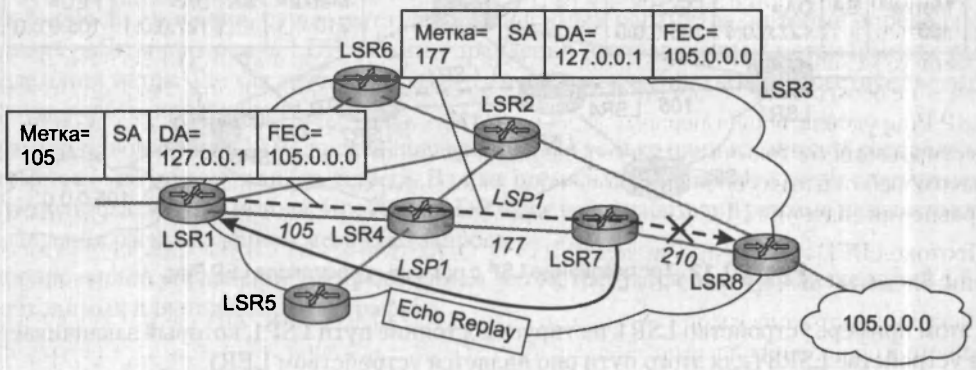


Рис. 20.13. Тестирование неисправного пути LSP с помощью протокола LSP Ping

В этой ситуации LSR7 не может отправить MPLS-кадр по назначению, как того требует метка 177, а отбрасывает заголовок MPLS и старается обработать кадр как IP-пакет. Как и в случае исправного пути, адрес 127.0.0.1 требует передачи пакета локальному стеку TCP/IP. Именно этого эффекта и добивались разработчики протокола LSP Ping, выбирая в качестве адреса назначения этот специальный адрес. Узел LSR7 обрабатывает сообщение *Echo Request* и отправляет сообщение *Echo Replay* узлу LSR1 с информацией об обнаруженной ошибке.

Трассировка путей LSP

При неисправном состоянии какого-то отрезка пути LSP сообщение об ошибке не всегда может быть отправлено промежуточным устройством LSP. Возможна и такая ситуация, когда ответ на запрос *Echo Request* просто не приходит — сеть «молчит», например, потому что отказал промежуточный узел. Для того чтобы локализовать отказавший элемент сети (узел или соединение), протокол LSP Ping может работать в режиме трассировки пути LSP. Этот режим аналогичен режиму работы утилиты traceroute стека TCP/IP и в нем используется тот же механизм, заключающийся в посылке серии сообщений *Echo Request*

с монотонно возрастающим от 1 значением поля TTL. Разница состоит в том, что это поле указывается не в IP-пакете, как при использовании IP-утилиты traceroute, а в заголовке MPLS (который также имеет поле TTL).

Дальнейшее поведение протокола LSP Ping в режиме трассировки очевидно — MPLS-кадр с нулевым значением TTL передается «наверх» протоколу LSP Ping того промежуточного узла, который после вычитания единицы из значения этого поля получил нулевой результат. Протокол реагирует на такую ситуацию отправкой сообщения *Echo Replay* начальному узлу тестируемого пути.

Протокол двунаправленного обнаружения ошибок продвижения

Протокол **двунаправленного обнаружения ошибок продвижения** (Biderectional Forwarding Detection, BFD) разработан как «облегченная» альтернатива протоколу LSP Ping для постоянного мониторинга состояния пути LSP. Такой постоянный мониторинг требуется, например, в тех случаях, когда основной путь защищен резервным путем и необходим какой-то механизм, который, с одной стороны, может быстро выявить отказ пути, а с другой — не перегружает сеть тестовыми сообщениями и трудоемкими проверками. Протокол LSP Ping удовлетворяет первому условию, то есть может использоваться для постоянного тестирования состояния пути путем периодической отправки сообщений *Echo Requst*. Однако обработка этих сообщений конечным узлом пути довольно трудоемка, так как требует сравнения значения FEC в каждом пришедшем запросе со значением из базы данных.

Протокол BFD гораздо проще, чем LSP Ping. Однако он не способен локализовать отказавший элемент сети, а только показывает, работоспособен некоторый путь LSP или нет.

Название протокола говорит о том, что он проверяет состояние соединения между двумя узлами в обоих направлениях. Так как пути MPLS однонаправленные, то для работы протокола BFD необходима пара путей LSP, соединяющих два узла в обоих направлениях.

Каждый из двух конечных узлов, на которых для мониторинга определенного пути LSP развернут протокол BFD, периодически посылает по этому пути сообщения *Hello*. Получение сообщений Hello от соседа означает работоспособность пути в одном определенном направлении. Неполучение сообщения *Hello* в течение определенного времени означает отказ пути в этом направлении, что и фиксирует протокол BFD. Информацию об отказе пути могут немедленно использовать другие протоколы стека MPLS, например рассматриваемые далее протоколы защиты пути.

Протокол BFD посылает сообщения *Hello* в UDP-сообщениях, которые, в свою очередь, упаковываются в IP-пакеты и снабжаются заголовками MPLS. Протокол BFD может использоваться не только для мониторинга путей MPLS, он разработан как универсальный протокол тестирования двунаправленных соединений. Обычно для инициализации сеанса BFD служит протокол LSP Ping, который переносит по пути идентификаторы сеанса BFD.

Инжиниринг трафика в MPLS

Технология MPLS поддерживает технику инжиниринга трафика, описанную в главе 7. В этом случае используются модифицированные протоколы сигнализации и маршрутиза-

ции, имеющие приставку TE (Traffic Engineering – инжиниринг трафика). В целом такой вариант MPLS получил название MPLS TE.

В технологии MPLS TE пути LSP называют **TE-туннелями**. TE-туннели не прокладываются распределенным способом вдоль путей, находимых обычными протоколами маршрутизации независимо в каждом отдельном устройстве LSR. Вместо этого TE-туннели прокладываются в соответствии с техникой маршрутизации от источника, когда централизованно задаются промежуточные узлы маршрута. В этом отношении TE-туннели подобны PVC-каналам в технологиях ATM и Frame Relay. Инициатором задания маршрута для TE-туннеля выступает начальный узел туннеля, а рассчитываться такой маршрут может как этим же начальным узлом, так и внешней по отношению к сети программной системой или администратором.

MPLS TE поддерживает туннели двух типов:

- **строгий TE-туннель** определяет все промежуточные узлы между двумя пограничными устройствами;
- **свободный TE-туннель** определяет только часть промежуточных узлов от одного пограничного устройства до другого, а остальные промежуточные узлы выбираются устройством LSR самостоятельно.

На рис. 20.14 показаны оба типа туннелей.

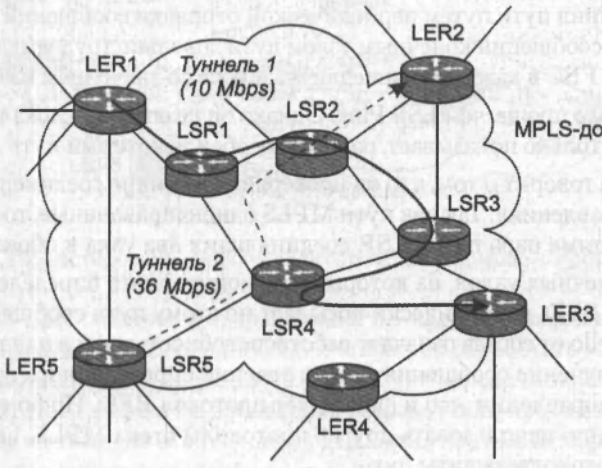


Рис. 20.14. Два типа TE-туннелей в технологии MPLS

Туннель 1 является примером строгого туннеля, при его задании внешняя система (или администратор сети) указала как начальный и конечный узлы туннеля, так и все промежуточные узлы, то есть последовательность IP-адресов для устройств LER1, LSR1, LSR2, LSR3, LER3. Таким образом, внешняя система решила задачу инжиниринга трафика, выбрав путь с достаточной неиспользуемой пропускной способностью. При установлении туннеля 1 задается не только последовательность LSR, но и требуемая пропускная способность пути. Несмотря на то что выбор пути происходит в автономном режиме, все устройства сети вдоль туннеля 1 проверяют, действительно ли они обладают запрошенной неиспользуемой пропускной способностью, и только в случае положительного ответа туннель прокладывается.

При прокладке туннеля 2 (свободного) администратор задает только начальный и конечный узлы туннеля, то есть устройства LER5 и LER2. Промежуточные устройства LSR4 и LSR2 находятся автоматически начальным узлом туннеля 2, то есть устройством LER5, а затем с помощью сигнального протокола устройство LER5 сообщает этим и конечному устройствам о необходимости прокладки туннеля.

Независимо от типа туннеля он всегда обладает таким параметром, как резервируемая пропускная способность. В нашем примере туннель 1 резервирует для трафика 10 Мбит/с, а туннель 2 — 36 Мбит/с. Эти значения определяются администратором, и технология MPLS TE никак не влияет на их выбор, она только реализует запрошенное резервирование. Чаще всего администратор оценивает резервируемую для туннеля пропускную способность на основании измерений трафика в сети, тенденций изменения трафика, а также собственной интуиции. Некоторые реализации MPLS TE позволяют затем автоматически корректировать величину зарезервированной пропускной способности на основании автоматических измерений реальной интенсивности трафика, проходящего через туннель.

Однако сама по себе прокладка в MPLS-сети TE-туннеля еще не означает передачи по нему трафика. Она означает только то, что в сети действительно существует возможность передачи трафика по туннелю со средней скоростью, не превышающей зарезервированное значение. Для того чтобы данные были переданы по туннелю, администратору предстоит еще одна ручная процедура — задание для начального устройства туннеля условий, определяющих, какие именно пакеты должны передаваться по туннелю. Условия могут быть чрезвычайно разнообразными, так, в качестве признаков агрегированного потока, который должен передаваться по туннелю, могут выступать все традиционные признаки: IP-адрес назначения и источника, тип протокола, номера TCP- и UDP-портов, номер интерфейса входящего трафика, значения приоритета в протоколах DSCP и IP и т. д.

Таким образом, устройство LER должно сначала провести *классификацию трафика*, затем выполнить *профилирование*, удостоверившись, что средняя скорость потока не превышает зарезервированную, и наконец, начать *маркировать* пакеты, используя начальную метку TE-туннеля, чтобы передавать трафик через сеть с помощью техники MPLS. В этом случае расчеты, выполненные на этапе выбора пути для туннеля, дадут нужный результат — баланс ресурсов сети при соблюдении средней скорости для каждого потока.

Однако мы еще не рассмотрели специфический набор протоколов, которые устройства LER и LSR сети используют для прокладки свободных туннелей или проверки работоспособности созданных администратором строгих туннелей.

Для выбора и проверки путей через туннели в технологи MPLS TE используются расширения протоколов маршрутизации, работающих на основе алгоритма состояния связей. Сегодня такие расширения стандартизованы для протоколов OSPF и IS-IS. Для решения задачи TE в протоколы OSPF и IS-IS включены новые типы объявлений, обеспечивающие распространение по сети информации о номинальной и незарезервированной (доступной для TE-потоков) величинах пропускной способности каждой связи. Таким образом, ребра результирующего графа сети, создаваемого в топологической базе каждого устройства LER или LSR, маркируются этими двумя дополнительными параметрами. Располагая таким графом, а также параметрами потоков, для которых нужно определить TE-пути, устройство LER может найти рациональное решение, удовлетворяющее одному из сформулированных в главе 7 ограничений на использование ресурсов сети. Чаще всего решение ищется по наиболее простому критерию, который состоит в минимизации максимального значения

коэффициента использования вдоль выбранного пути, то есть критерием оптимизации пути является значение min (max Ki) для всех возможных путей.

В общем случае администратору необходимо проложить несколько туннелей для различных агрегированных потоков. С целью упрощения задачи оптимизации выбор путей для этих туннелей обычно осуществляется по очереди, причем администратор определяет очередность на основе своей интуиции. Очевидно, что поиск TE-путей по очереди снижает качество решения — при одновременном рассмотрении всех потоков в принципе можно было бы добиваться более рациональной загрузки ресурсов.

ПРИМЕР

В примере, показанном на рис. 20.15, ограничением является максимально допустимое значение коэффициента использования ресурсов, равное 0,65. В варианте 1 решение было найдено при очередности рассмотрения потоков 1, 2, 3. Для первого потока был выбран путь *A-B-C*, так как в этом случае он, с одной стороны, удовлетворяет ограничению (все ресурсы вдоль пути — каналы *A-B*, *A-C* и соответствующие интерфейсы маршрутизаторов оказываются загруженными на 50/155 = 0,32), а с другой — обладает минимальной метрикой (65 + 65 = = 130). Для второго потока также был выбран путь *A-B-C*, так как и в этом случае ограничение удовлетворяется — результирующий коэффициент использования оказывается равным 50 + 40/155 = 0,58. Третий поток направляется по пути *A-D-E-C* и загружает ресурсы каналов *A-D*, *D-E* и *E-C* на 0,3. Решение 1 можно назвать удовлетворительным, так как коэффициент использования любого ресурса в сети не превышает 0,58.

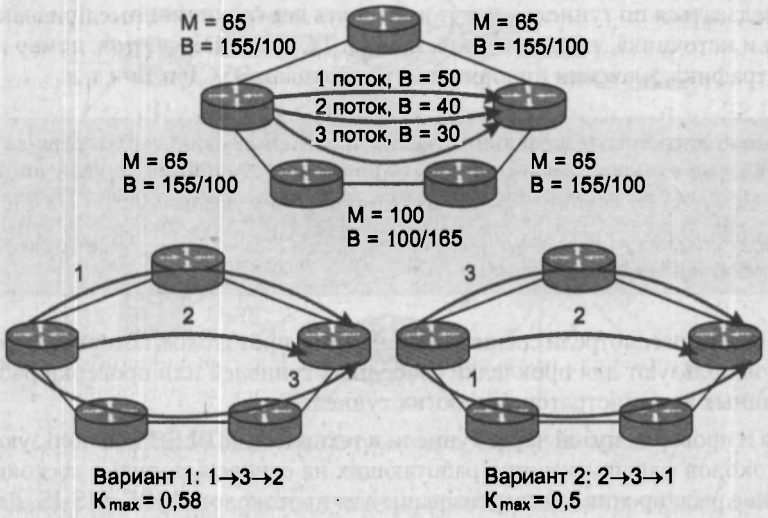


Рис. 20.15. Зависимость качества решения задачи TE от очередности выбора туннелей

Однако существует лучший способ, представленный в варианте 2. Здесь потоки 2 и 3 были направлены по верхнему пути *A-B-C*, а поток 1 — по нижнему пути *A-D-E-C*. Ресурсы верхнего пути оказываются загруженными на 0,45, а нижнего — на 0,5, то есть налицо более равномерная загрузка ресурсов, а максимальный коэффициент использования всех ресурсов сети не превышает 0,5. Этот вариант может быть получен при одновременном рассмотрении всех трех потоков с учетом ограничения min (max Ki) или же при рассмотрении потоков по очереди в последовательности 2, 3, 1.

Несмотря на не оптимальность качества решения, в производимом сегодня оборудовании применяется вариант технологии MPLS TE с последовательным рассмотрением потоков. Он проще в реализации и ближе к стандартным для протоколов OSPF и IS-IS процедурам нахождения кратчайшего пути для одной сети назначения (в отсутствие ограничений найденное решение для набора кратчайших путей не зависит от последовательности учета сетей, для которых производился поиск). Кроме того, при изменении ситуации — появлении новых потоков или изменении интенсивности существующих — найти путь удается только для одного потока.

Возможен также подход, в котором внешняя по отношению к сети вычислительная система, работающая в автономном режиме, определяет оптимальное решение для набора потоков. Это может быть достаточно сложная система, которая включает подсистему имитационного моделирования, способную учесть не только средние интенсивности потоков, но и их пульсации и оценить не только загрузку ресурсов, но и результирующие параметры QoS — задержки, потери и т. п. После нахождения оптимального решения его можно модифицировать уже в оперативном режиме поочередного поиска путей.

В технологии MPLS TE информация о найденном рациональном пути используется полностью, то есть запоминаются IP-адреса источника, всех транзитных маршрутизаторов и конечного узла. Поэтому достаточно, чтобы поиском путей занимались только пограничные устройства сети (LER), а промежуточные устройства (LSR) лишь поставляли им информацию о текущем состоянии резервирования пропускной способности каналов.

После нахождения пути независимо от того, найден он был устройством LER или администратором, его необходимо зафиксировать. Для этого в MPLS TE используется расширение уже рассмотренного нами протокола резервирования ресурсов (RSVP), который часто в этом случае называют протоколом **RSVP TE**. Сообщения RSVP TE передаются от одного устройства LSR другому в соответствии с данными о найденных IP-адресах маршрута. При установлении нового пути в сигнальном сообщении наряду с последовательностью адресов пути указывается также и резервируемая пропускная способность. Каждое устройство LSR, получив такое сообщение, вычитает запрашиваемую пропускную способность из пула свободной пропускной способности соответствующего интерфейса, а затем объявляет остаток в сообщениях протокола маршрутизации, например CSPF.

В заключение рассмотрим вопрос отношения технологий MPLS TE и QoS. Как видно из описания, основной целью MPLS TE является использование возможностей MPLS для достижения внутренней цели поставщика услуг, а именно сбалансированной загрузки всех ресурсов своей сети. Однако при этом также создается основа для предоставления транспортных услуг с гарантированными параметрами QoS, так как трафик по TE-туннелям передается при соблюдении некоторого максимального уровня коэффициента использования ресурсов. Как мы знаем из материала главы 7, коэффициент использования ресурсов оказывает решающее влияние на процесс образования очереди, так что потоки, передаваемые по TE-туннелям, передаются с некоторым гарантированным уровнем QoS.

Для того чтобы обеспечить разные параметры QoS для разных классов трафика, поставщику услуг необходимо для каждого класса трафика установить в сети отдельную систему туннелей. При этом для чувствительного к задержкам класса трафика требуется выполнить резервирование таким образом, чтобы максимальный коэффициент использования ресурсов туннеля находился в диапазоне 0,2–0,3, иначе задержки пакетов и их вариации выйдут за допустимые пределы.

Отказоустойчивость путей MPLS

Общая характеристика

MPLS поддерживает несколько механизмов обеспечения отказоустойчивости, или в терминах SDH — механизмов *автоматического защитного переключения* маршрута в случае отказа какого-либо элемента сети: интерфейса LSR, линии связи или LSR в целом.

В том случае, когда путь устанавливается с помощью протокола LDP, существует единственная возможность защиты пути — его восстановление с помощью распределенного механизма нахождения нового пути средствами протоколов маршрутизации. Это абсолютно тот же механизм, который используется в IP-сетях при отказе линии или маршрутизатора. Время восстановления пути зависит от применяемого протокола маршрутизации и сложности топологии сети, обычно это десятки секунд или несколько минут.

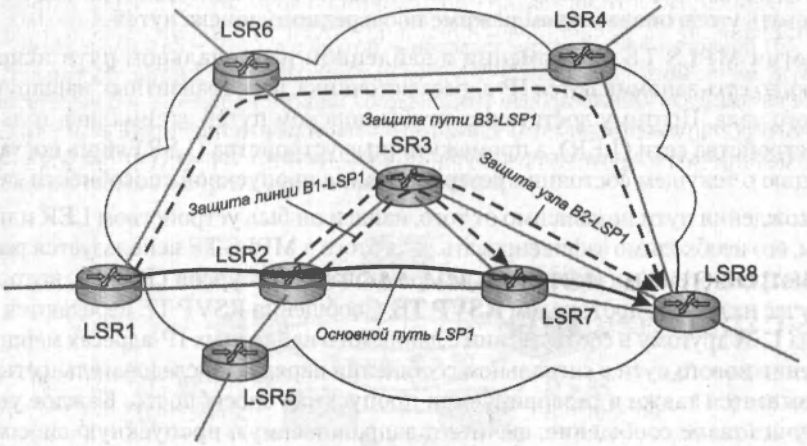


Рис. 20.16. Защитные механизмы MPLS

В том случае, когда путь является TE-туннелем, в технологии MPLS разработано несколько механизмов его восстановления. Эти механизмы иллюстрирует рис. 20.16, на котором показан основной путь LSP1, соединяющий устройства LSR1 и LSR8. Будем считать, что путь LSP1 является TE-туннелем.

- *Восстановление пути его начальным узлом.* Традиционное (с помощью протокола маршрутизации) повторное нахождение нового пути, обходящего отказавший элемент сети. Отличие от восстановления пути LDP заключается только в том, что прокладкой нового пути занимается лишь один узел сети, а именно начальный узел пути. В нашем примере это узел LSR1.
- *Защита линии.* Такая защита организуется между двумя устройствами LSR, непосредственно соединенными линией связи. Обходной маршрут находится заранее, до отказа линии, и заранее прокладывается между этими устройствами таким образом, чтобы обойти линию связи в случае ее отказа. В нашем примере такой вариант защиты установлен для линии, соединяющей узлы LSR2 и LSR7. Обходной путь B1-LSP1 проложен через узел LSR3. Защита линии является временной мерой, так как параллельно с началом использования обходного пути начальный узел основного пути начинает

процедуру его восстановления с помощью протокола маршрутизации. После восстановления основного пути использование обходного пути прекращается. Временная защита линии не гарантирует TE-туннелю требуемой пропускной способности. Механизм защиты линии работает очень быстро, обычно время переключения не превосходит 50 мс, то есть сравнимо со временем переключения сетей SDH, которые всегда выступают в этой области в качестве эталона. Поэтому механизм защиты линии называют быстрой перемаршрутизацией (fats re-route).

- *Защита узла.* Этот механизм очень похож на механизм защиты линии, но отличается тем, что обходной путь прокладывается так, чтобы обойти отказавшее устройство LSR (в нашем примере на рисунке это устройство LSR7). Все остальные характеристики аналогичны характеристикам защиты линии; механизм защиты узла тоже относится к механизмам быстрой перемаршрутизации и тоже является временной мерой.
- *Защита пути.* В дополнение к основному пути в сети прокладывается путь, связывающий те же конечные устройства, но проходящий по возможности через устройства LSR и линии связи, не встречающиеся в основном пути (на рисунке это резервный путь B3-LSP1). Данный механизм самый универсальный, но он работает медленнее, чем механизмы защиты линии и узла.

Для быстрого обнаружения отказа основного пути или его части могут использоваться различные механизмы и протоколы: сообщения *Hello* протокола RSVP, протокол LSP Ping или BFD.

Использование иерархии меток для быстрой защиты

Рассмотрим работу быстрых механизмов защиты на примере защиты линии, представленной на рис. 20.17. Пусть для защиты линии LSR2-LSR7 в сети проложен обходной путь B-LSP1. На основном пути LSP1 для продвижения кадров используется последовательность меток 15, 17 и 21. На первом участке обходного пути B-LSP1 используется метка 7, на втором — метка 8.

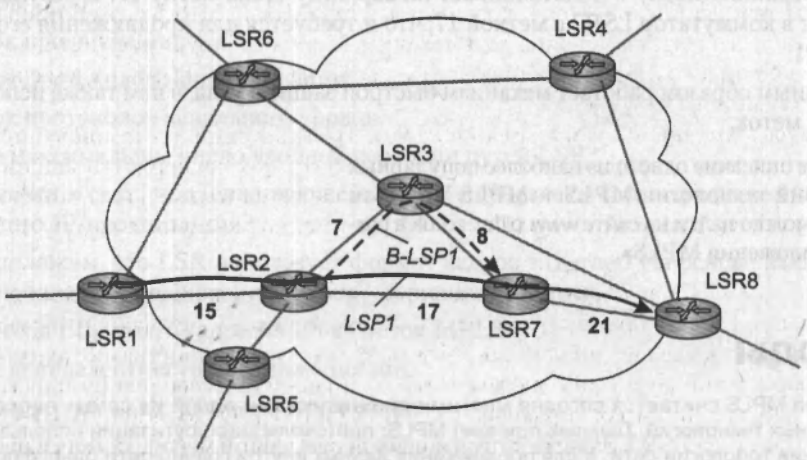


Рис. 20.17. Распределение меток для основного пути и обходного пути защиты линии

При отказе линии LSR2-LSR7 устройство LSR2 начинает направлять кадры, поступающие по пути LSP1, в обходной путь B-LSP1 (рис. 20.18). Однако если при этом поменять метку 15 на метку 7, как того требует обычная логика коммутации меток, то кадр придет в устройство LSR7 с меткой 8 (ее установит устройство LSR3), которая не соответствует значению метки 17, используемой в устройстве LSR7 для передачи кадров по пути LSP1.

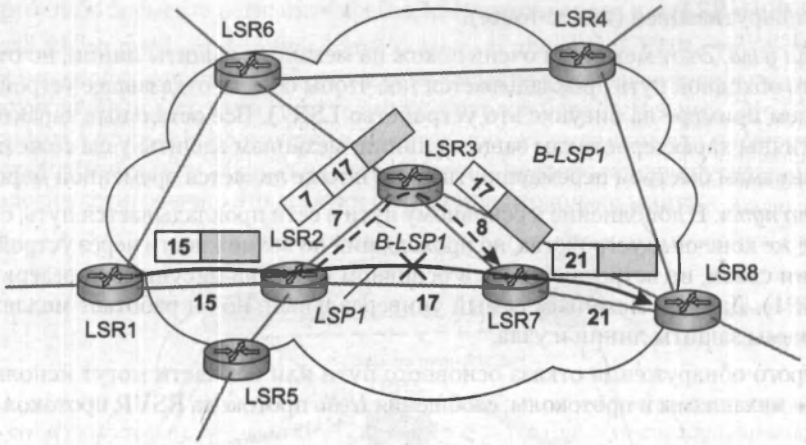


Рис. 20.18. Передачи кадров по обходному пути

Для того чтобы устройство LSR7 работало при переходе на обходной путь точно так же, как и при нормальной работе основного пути, в технике быстрой защиты применяется иерархия меток. Для этого устройство LSR2, которое реализует механизм защиты линии, заменяет метку 15 в пришедшем пакете меткой 17, как если бы линия LSR2-LSR7 не отказывала. Затем устройство LSR2 проталкивает метку первого уровня в стек, а на вершину стека помещает метку 7, которая нужна для продвижения кадра по обходному пути.

Устройство LSR3 является предпоследним устройством обходного пути. Поэтому оно удаляет верхнюю метку 7 и выталкивает на вершину стека метку 17. В результате кадр поступает в коммутатор LSR7 с меткой 17, что и требуется для продвижения его далее по пути LSP1.

Аналогичным образом работает механизм быстрой защиты узла, в нем также используется иерархия меток.

⇨ Подробное описание одного из наиболее популярных приложений технологии MPLS – MPLS VPN 3-го уровня – можно найти на сайте www.olifer.co.uk в разделе «Приложения MPLS».

Выводы

Технология MPLS считается сегодня многими специалистами одной из самых перспективных транспортных технологий. Главный принцип MPLS: протоколы маршрутизации используются для определения топологии сети, а для продвижения данных внутри границ сети одного поставщика услуг применяется техника виртуальных каналов.

Объединение техники виртуальных каналов с функциональностью стека TCP/IP происходит за счет того, что одно и то же сетевое устройство, называемое коммутирующим по меткам маршрутизатором (LSR), выполняет функции как IP-маршрутизатора, так и коммутатора виртуальных каналов.

Кадры MPLS имеют заголовки двух типов:

- внешний заголовок одной из технологий канального уровня, например Ethernet или PPP;
- заголовок-прокладка с полем метки и некоторыми другими полями, относящимися собственно к технологии MPLS.

MPLS поддерживает иерархию путей за счет применения техники стека меток. При этом число уровней иерархии не ограничено.

Протокол LDP позволяет автоматически назначать метки для вновь прокладываемого пути LSP. Маршрут для этого пути выбирается на основании работы стандартных протоколов маршрутизации.

Для тестирования состояния пути LSP в технологии MPLS разработан протокол LSP Ping, работа которого во многом похожа на работу утилиты ping стека TCP/IP. Мониторинг состояния пути LSP можно выполнять с помощью протокола BFD.

Существует несколько механизмов отказоустойчивости в сетях MPLS:

- восстановление пути его начальным узлом;
- защита линии;
- защита узла;
- защита пути.

Технология MPLS поддерживает инжиниринг трафика. Для этого применяются специальные версии протоколов маршрутизации, такие как OSPF TE и IS-IS TE, которые учитывают свободную пропускную способность каждой линии связи сети.

Автоматическое установление найденного в соответствии с задачами инжиниринга трафика пути осуществляется специальной версией протокола RSVP, которая имеет название RSVP TE.

Вопросы и задания

1. Технология MPLS является гибридом технологий:

   а) IP и IPX; б) IP и OSPF; в) IP и технологии виртуальных каналов.

2. Какие функциональные модули IP-маршрутизатора используются в LSR? Варианты ответов:

   а) блок продвижения;

   б) блок протоколов маршрутизации;

   в) блок протоколов канального уровня.

3. Какое максимальное число уровней иерархии путей LSP?

4. Можно ли в сети, поддерживающей MPLS, передавать часть трафика посредством обычного IP-продвижения?

5. Предположим, что LSR использует формат кадров Ethernet. На основе каких адресов LSR выполняет продвижение кадров? Варианты ответов:

   а) адресов Ethernet; б) адресов IP; в) меток MPLS.

6. Класс эквивалентности продвижения это:

   а) набор путей LSP с равными метриками;

   б) набор путей к одному и тому же выходному устройству LER;

   в) группа IP-пакетов, имеющих одни и те же требования к условиям транспортировки.

7. Что является аналогом туннелей MPLS TE в технологии ATM? Варианты ответов:
   а) постоянные виртуальные каналы;
   б) коммутируемые виртуальные каналы;
   в) иерархические соединения.
8. Протокол LDP позволяет автоматически проложить пути LSP, причем маршруты для них:
   а) определяются стандартной таблицей маршрутизации;
   б) определяются с помощью техники инжиниринга трафика;
   в) учитывают свободную пропускную способность линий связи.
9. Какой из вариантов управления распределением меток протоколом LDP называется упорядоченным? Варианты ответов:
   а) метка назначается по запросу от вышележащего устройства LSR;
   б) метка не назначается устройством LSR до тех пор, пока оно не получит метку от нижележащего устройства;
   в) метка назначается без запроса.
10. Зачем в сообщении *Echo Request* протокола LSP Ping в качестве IP-адреса назначения используется адрес обратной петли 127.0.0.1? Варианты ответов:
    а) для тестирования стека протоколов TCP/IP каждого промежуточного устройства LSR;
    б) этот адрес выбран произвольно и ни на что не влияет, потому что сообщение передается на основе меток MPLS;
    в) для передачи сообщения стеку протоколов TCP/IP узла тестируемого пути, после которого путь поврежден.
11. Протокол BFD отличается от протокола LSP Ping следующими свойствами:
    а) не может тестировать многодоменные пути;
    б) проще в реализации;
    в) не способен локализовать неисправности.
12. Какие узлы пути задаются при описании свободного TE-пути?
    а) только конечный; б) начальный и конечный; в) часть промежуточных узлов.
13. Какие механизмы отказоустойчивости путей MPLS являются самыми быстрыми? Варианты ответов:
    а) восстановление пути его начальным узлом;
    б) защита узла;
    в) защита линии;
    г) защита пути.

ГЛАВА 21 Ethernet операторского класса

Ethernet операторского класса (Carrier Ethernet, или Carrier Grade Ethernet) — это сравнительно новый термин, под которым скрывается целый спектр различных технологий.

В наиболее широком смысле под Ethernet операторского класса понимают как услуги Ethernet, которые операторы связи предоставляют в глобальном масштабе, так и технологии, на основе которых эти услуги организуются. В эти технологии входит усовершенствованная версия Ethernet, а также MPLS и технологии первичных сетей, такие как SDH, OTN и DWDM.

В этой главе мы рассмотрим наиболее популярные технологии, входящие в семейство Ethernet операторского класса, а также формализованное описание услуг Ethernet операторского класса.

Обзор версий Ethernet операторского класса

Движущие силы экспансии Ethernet

Как мы знаем, классическая технология Ethernet разрабатывалась исключительно как технология локальных сетей, и до недавнего времени сети этого класса и были единственной областью ее применения. Однако бесспорный успех Ethernet в локальных сетях, где она вытеснила все остальные технологии, привел к напрашивающейся идее об использовании этой технологии и в глобальных сетях (которые по большей части являются операторскими).

Потенциальных преимуществ от экспансии Ethernet за пределы локальных сетей несколько.

Для пользователей важно появление Ethernet *как услуги глобальных сетей*. Эта услуга может у разных провайдеров называться по-разному — Carrier Ethernet, Ethernet VPN, VPLS, ELINE или ELAN — суть от этого не меняется: пользователи получают возможность соединения своих территориально рассредоточенных сетей так же, как они привыкли в своих офисных сетях, то есть на уровне коммутаторов Ethernet и без привлечения протокола IP. При этом пользователи имеют дело с хорошо изученной технологией на интерфейсах, соединяющих их пограничное оборудование с пограничным оборудованием провайдера. Кроме того, при соединении сетей на канальном уровне пользователи свободны в IP-адресации своих сетей, так как при передаче трафика между сетями пользователей услуги Ethernet операторского класса провайдер не применяет IP-адреса. Таким образом, можно, например, назначить адреса одной и той же IP-подсети для всех сетей пользователей или же задействовать частные IP-адреса. Это общее свойство услуг *VPN канального уровня*, но сегодня такая услуга практически всегда выглядит как услуга с интерфейсом Ethernet.

Очень полезным свойством является также мобильность сетей пользователей; так, при помещении какой-либо сети пользователя в центр данных провайдера (то есть при хостинге сети Ethernet) ее IP-адреса могут оставаться теми же, что и были прежде, когда эта сеть была составной частью корпоративной сети пользователя.

Для провайдеров Ethernet операторского класса важна и как популярная услуга, и как *внутренняя транспортная технология канального уровня*. В последнем случае эта технология может использоваться для реализации глобальных услуг Ethernet или же для создания надежных, быстрых и контролируемых соединений между маршрутизаторами.

Привлекательность Ethernet как внутренней транспортной технологии для операторов связи объясняется относительно низкой стоимостью оборудования Ethernet. Порты Ethernet всегда обладали самой низкой стоимостью по сравнению с портами любой другой технологии (естественно, с учетом скорости передачи данных портом). Низкая стоимость изначально была результатом простоты технологии Etherhet, которая предлагает только минимальный набор функций по передаче кадров в режиме доставки по возможности (с максимальными усилиями), не поддерживая ни контроль над маршрутами трафика, ни мониторинг работоспособности соединения между узлами. Низкая стоимость оборудования Ethernet при удовлетворительной функциональности привела к доминированию Ethernet на рынке оборудования для локальных сетей, ну а далее начал работать механизм

положительной обратной связи: хорошие продажи — массовое производство — еще более низкая стоимость и т. д.

Стремление к унификации также относятся к силам, ведущим к экспансии Ethernet в глобальные сети. Сетевой уровень уже давно демонстрирует однородность благодаря доминированию протокола IP, и перспектива получить однородный канальный уровень в виде Ethernet выглядит очень заманчивой.

Однако все это относится к области желаний, а как обстоит дело с возможностями? Готова ли технология Ethernet к новой миссии? Ответ очевиден — в своем классическом виде технологии локальной сети не готова. Для того чтобы успешно работать в сетях операторов связи, технология и воплощающее ее оборудование должны обладать определенным набором характеристик, среди которых, в первую очередь, нужно отметить надежность, отказоустойчивость, масштабируемость и управляемость. Эталоном такой технологии может служить технология SDH, которая долгие годы использовалась (и все еще используется) как становой хребет сетей операторов связи, соединяя своими каналами маршрутизаторы, телефонные станции и любое другое оборудование провайдера. MPLS также может выступать в качестве эталона технологии операторского класса, ее основные свойства, описываемые в главе 20, позволяют сделать такой вывод.

Для того чтобы соперничать с SDH или MPLS, превратившись в технологию операторского класса, Ethernet надо улучшить свою функциональность, при этом наиболее важным является решение двух задач:

- Эксплуатационные и административные характеристики должны поддерживаться протоколами администрирования и обеспечивать мониторинг состояния соединений, а также локализацию и устранение неисправностей. Эти характеристики необходимы для успешного применения Ethetrnet в качестве внутренней транспортной технологии операторов связи.
- Должна быть обеспечена изоляция адресных пространств сети Ethernet провайдера от адресных пространств сетей Ethernet пользователей. Как вы знаете, пространство MAC-адресов Ethernet является плоским, так что если сеть Ethernet провайдера соединить непосредственно (а не через маршрутизатор) с сетями Ethernet пользователей, то всем коммутаторам сети Ethernet провайдера придется иметь дело с MAC-адресами пользовательского оборудования, а у крупного провайдера их может насчитываться сотни тысяч. Здесь требуется какое-то принципиально другое решение, иначе провайдер не сможет оказывать услуги частных виртуальных сетей Ethernet, строя их на собственном оборудовании Ethernet.

Разные «лица» Ethernet

Как мы увидим далее, разработчики технологии Ethernet на пути превращения ее в технологию операторского класса пытаются решить обе задачи. Однако из-за того, что такая работа начата сравнительно недавно, для оказания глобальных услуг Ethernet первыми в сетях операторов связи стали применяться технологии, отличные от Ethernet. И только в последнее время к ним присоединилась собственно технология Ethernet.

Ситуацию в области Ethernet операторского класса иллюстрирует рис. 21.1. Он показывает, что независимо от внутренней реализации *для пользователя глобальная услуга Ethernet всегда предоставляется с помощью набора стандартных интерфейсов Ethernet (Ethernet UNI) на каналах доступа к сети провайдера.*

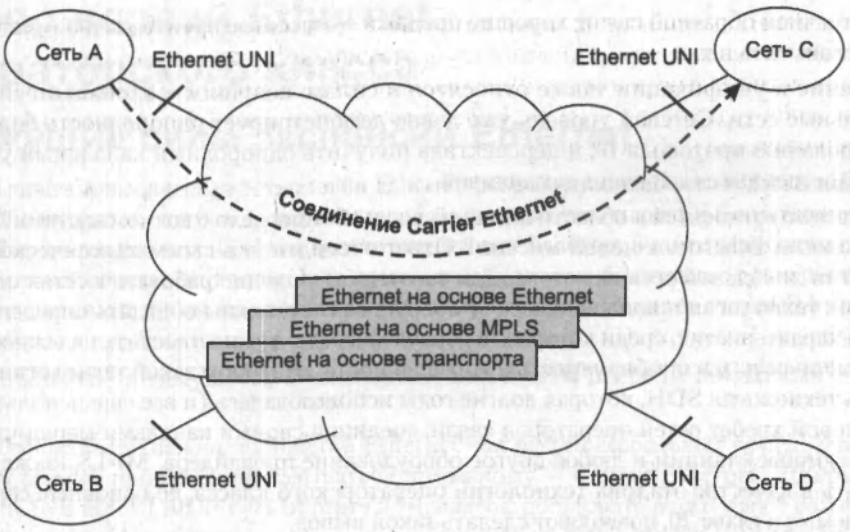


Рис. 21.1. Различные варианты реализации услуги Carrier Ethernet

Эти интерфейсы поддерживают одну из спецификаций Ethernet физического уровня, например 100Base-FX или 1000Base-LX, а также стандартные кадры Ethernet. Кроме того, существует некоторое описание услуги, которое определяет ее основные параметры, такие как топологию взаимодействия сетей пользователей (например, двухточечную, как показано на рисунке, звездообразную или полносвязную), пропускную способность логического соединения или же гарантированный уровень качества обслуживания кадров.

Однако если внешне услуги Ethernet операторского класса у разных провайдеров выглядят более-менее однотипно, внутренняя организация такой услуги в пределах сети провайдера может отличаться значительно.

Сегодня можно выделить три основных варианта подобной организации в зависимости от используемой внутренней транспортной технологии.

- **Ethernet поверх MPLS** (Ethernet over MPLS, EoMPLS). В этом случае MPLS-туннели (с некоторой надстройкой) используются как основной транспортный механизм провайдера, позволяющий эмулировать услугу Etheret для клиентов. Такие свойства MPLS, как поддержка детерминированных маршрутов, наличие механизма быстрой перемаршрутизации, обеспечивающего быстрое (сравнимое с SDH) переключение с основного маршрута на резервный, развитые средства контроля работоспособности соединений, сделали эту технологию весьма привлекательной для операторов связи. Кроме того, MPLS — это весьма зрелая технология с более чем 10-летней историей; она используется сегодня в магистральных сетях очень многих крупных провайдеров связи для различных целей, так что ее надежность и эффективность проверены практикой. Группа IETF, занимающаяся разработкой стандартов MPLS, выпустила несколько документов RFC, описывающих детали процесса эмуляции Ethernet с помощью этой технологии. Сегодня данный подход является одним из самых распространенных при реализации услуги Ethernet VPN в сетях операторов связи.
- **Ethernet поверх Ethernet** (Ethernet over Ethernet), или **транспорт Ethernet операторского класса** (Carrier Ethernet Transport, CET). Этот вариант оказания глобальной

услуги Ethernet основан на использования в сети провайдера улучшенной версии Ethernet. Несколько названий этого варианта свидетельствуют о его молодости, когда терминология еще не устоялась и специалистам и пользователям приходится в начале обсуждения тратить время на то, чтобы договориться о взаимно приемлемом употреблении названий и аббревиатур.

Усилия разработчиков технологии CET (в дальнейшем будем использовать эту наиболее краткую аббревиатуру) и услуг на ее основе стандартизует комитет 802 IEEE. Из-за молодости этого направления не все его стандарты еще приняты, но приверженцы Ethernet могут назвать его «истинной» технологией Carrier Ethernet, так как здесь технология Ethernet не только видна потребителям услуг извне, но и работает внутри сети провайдера. Название транспорт Ethernet операторского класса как раз и отражает тот факт, что Ethernet операторского класса функционирует как транспортная технология провайдера.

Для любой пакетной технологии непросто приблизиться к функциональности SDH, а для Ethernet это сделать сложнее, чем, скажем, для MPLS, так как Ethernet изначально была задумана как дейтаграммная технология с минимумом функций. Тем не менее прогресс в этой области наблюдается.

- **Ethernet поверх транспорта** (Ethernet over Transport, EOT). Это наиболее традиционный для оператора связи вариант организации, так как под транспортом здесь понимается транспорт, основанный на технике коммутации каналов, которая всегда использовалась для создания первичных сетей операторов, то есть транспорт PDH, SDH или OTN. Для того чтобы эмулировать услуги Ethernet, необходимы некоторые надстройки над базовыми стандартами этих технологий, стандартизацией таких надстроек занимается ITU-T.

Стандартизация Ethernet как услуги

Стандартизация Ethernet как услуги — это еще одно важное направление работ в области Ethernet операторского класса, так как разнообразие реализаций этой услуги неминуемо приводит к разнообразию понятий, терминов и т. п., что весьма нежелательно.

Работой по созданию технологически нейтральных спецификаций глобальной услуги Ethernet занимается организация под названием Metro Ethernet Forum (MEF).

Использование термина Metro в названии этой организации отражает начальную ситуацию развития Ethernet операторского класса, когда такие услуги предоставлялись в основном в масштабах города. Теперь же, когда технология Ethenet операторского класса стала применяться и в глобальных масштабах, название можно было бы и поменять, но оно уже стало настолько популярным, что такое переименование вряд ли случится.

Организация MEF разработала несколько спецификаций, которые позволяют потребителю и поставщику услуги разработать нужный вариант услуги Ethernet, используя терминологию и параметры, не зависящие от конкретной внутренней реализации этой услуги провайдером. Такой подход удобен, он позволяет потребителям не знать терминологии той технологии, которую использует поставщик, например MPLS или SDH, и в то же время сознательно выбирать нужный ему вариант услуги.

В MEF вводится три типа услуг виртуальных частных сетей Ethernet, которые отличаются топологией связей между сайтами пользователей. Для того чтобы формализовать

топологию связей, вводится понятие **виртуального соединения Ethernet** (Ethernet Virtual Circuit, EVC). Каждое соединение EVC связывает сайты пользователей в отдельную виртуальную частную сеть, объединяя сетевые интерфейсы пользователей (User Network Interface, UNI).

Соответственно, имеются три типа соединений EVC (рис. 21.2):

- «точка-точка» (двухточечная топология);
- «каждый с каждым» (полносвязная топология);
- «дерево» (древовидная топология).

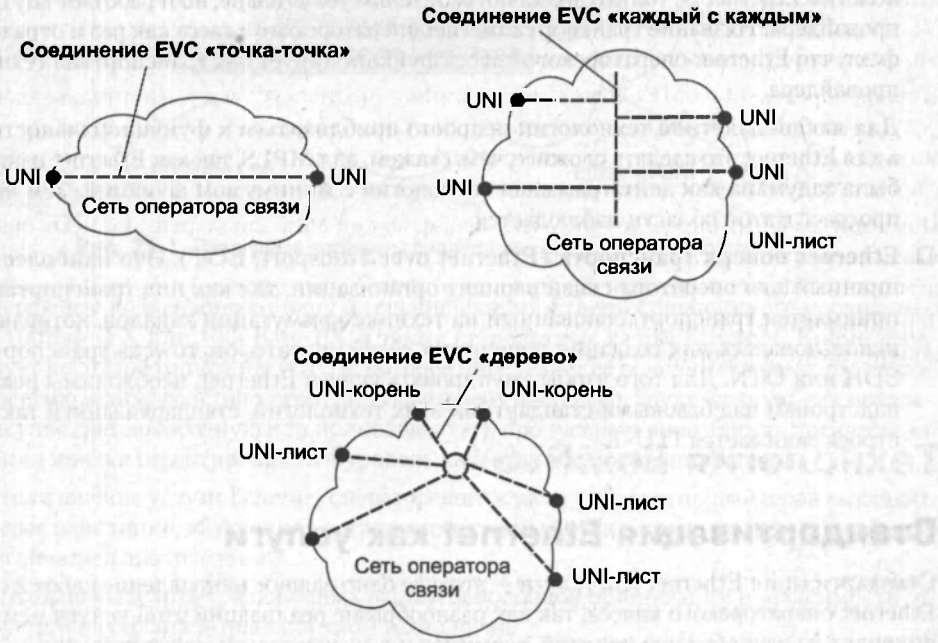


Рис. 21.2. Три типа услуг Ethernet

В зависимости от типа используемого соединения различаются и типы услуг:

- *E-LINE*. Эта услуга связывает только два пользовательских сайта через двухточечное EVC-соединение. Услуга E-LINE соответствует услуге выделенной линии.
- *E-LAN*. Эта услуга аналогична услуге локальной сети, так как она позволяет связать неограниченное число пользовательских сайтов таким образом, что каждый сайт может взаимодействовать с каждым. При этом соблюдается логика работы локальной сети — кадры Ethernet с неизученными и широковещательными MAC-адресами передаются всем сайтам, а кадры с изученными уникальными MAC-адресами — только тому сайту, в котором находится конечный узел с данным адресом.
- *E-TREE*. Спецификация этой услуги появилась позже других; в локальных сетях ей аналога нет. Пользовательские сайты делятся на корневые и листовые. Листовые сайты могут взаимодействовать только с корневыми, но не между собой. Корневые сайты могут взаимодействовать друг с другом.

Кроме того, в спецификациях MEF вводятся два варианта каждого типа услуги. В первом варианте пользовательский сайт определяется как сеть, подключенная к отдельному физическому интерфейсу UNI. Значения идентификаторов VLAN в пользовательских кадрах в расчет не принимаются. В названии этого варианта услуги к названию типа добавляется термин «частный» (private), например, для услуги типа E-LINE этот вариант называют частной линией Ethernet (Ethernet Private Line, EPL).

В другом варианте услуги к одному и тому же физическому интерфейсу UNI могут быть подключены различные пользовательские сайты. В этом случае они различаются по значению идентификатора VLAN. Другими словами, провайдер внутри своей сети сохраняет деление локальной сети на VLAN, сделанное пользователем. В варианте услуги с учетом VLAN добавляется название «виртуальная частная», например для услуги типа E-LINE это будет виртуальная частная линия Ethernet (Ethernet Virtual Private Line, EVPL).

В своих определениях MEF использует термины «частная услуга» и «виртуальная частная услуга» не совсем традиционным образом, так как оба типа услуги являются виртуальными частными в том смысле, что они предоставляются через логическое соединение в сети с коммутацией пакетов, а не через физический канал в сети с коммутацией каналов.

Помимо указанных определений услуг, спецификации MEF стандартизуют некоторые важные параметры услуг, например услуга может характеризоваться гарантированным уровнем пропускной способности соединения, а также гарантированными параметрами QoS.

Терминология MEF пока не получила широкого распространения. Во многих стандартах конкретных технологий по-прежнему употребляются собственные термины.

Технология EoMPLS

Псевдоканалы

Стандарты IETF описывают два типа услуг Ethernet операторского класса, которые строятся с помощью технологии MPLS: VPWS (Virtual Private Wire Service) и VPLS (Virtual Private LAN Service). Различие между этими услугами в том, что VPWS эмулирует соединение Ethernet с двухточечной топологией, то есть канал Ethernet, а VPLS эмулирует поведение локальной сети, то есть обеспечивает соединения с полносвязной топологией в стиле обычной локальной сети Ethernet.

Если использовать терминологию MEF, то услуга VPLS соответствует услуге E-LAN, а услуга VPWS – услуге E-LINE. При этом стандарты IETF описывают оба варианта услуг, как с принятием во внимание идентификаторов VLAN пользователя, так и без.

Обе услуги являются услугами MPLS VPN второго уровня (MPLS L2VPN), так как они позволяют предоставлять услуги VPN, взаимодействуя с пользовательскими сетями на втором уровне. В этом их отличие от услуг MPLS L3VPN, о которых рассказывается в главе 20.

Основным строительным элементом этих услуг являются так называемые **псевдоканалы**[1] (pseudowire), которые соединяют пограничные маршрутизаторы провайдера.

---

[1] Встречаются и другие русские переводы термина pseudowire, например эмулятор канала, эмулятор кабеля, псевдопровод.

На рис. 21.3 показано три таких псевдоканала, соединяющих между собой пограничные маршрутизаторы PE1–PE4.

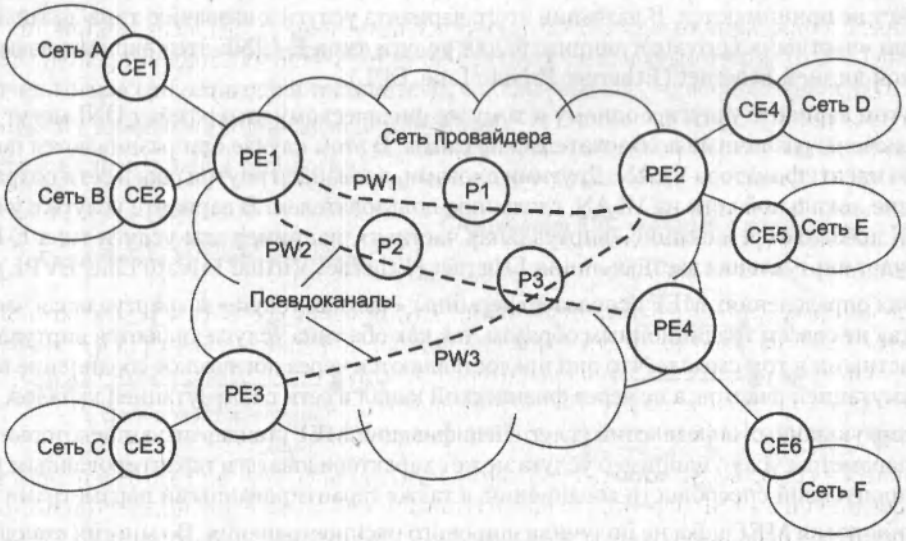


Рис. 21.3. Псевдоканалы в сети провайдера

Псевдоканалы представляют собой пути LSP второго уровня иерархии (называемого также внутренним уровнем), проложенным внутри LSP первого (внешнего) уровня. Обычно в качестве LSP первого уровня иерархии используются TE-туннели MPLS, так как они обладают такими дополнительными свойствами, которых нет у путей, проложенных с помощью протокола LDP. На рис. 21.3 пути LSP первого уровня не показаны, чтобы заострить внимание читателя на псевдоканалах.

Псевдоканалы — это логические транспортные соединения, физически они могут проходить через промежуточные магистральные маршрутизаторы, однако для них они прозрачны, то есть в нашем примере маршрутизаторы P1, P2 и P3 просто не замечают их существование в сети.

Однако псевдоканал — это не просто логическое соединение LSP второго уровня иерархии, согласно определению, данному в RFC 3985 (http://www.rfc-editor.org/rfc/rfc3985.txt), у псевдоканала есть более специфическое назначение.

> Псевдоканал — это механизм, который эмулирует существенные свойства какого-либо телекоммуникационного сервиса через сеть с коммуникацией пакетов.

Одним из вариантов применения псевдоканалов при эмуляции услуг Ethernet является передача псевдоканалом трафика одного пользовательского соединения, при этом псевдоканал эмулирует кабельное соединение между сетями пользователей. В примере на рис. 21.3 псевдоканал PW2 служит для организации соединения между сетями *A* и *F* через сеть провайдера. При этом кадры Ethernet, отправляемые сетью *A* в сеть *F*, инкапсулиру-

ются пограничным маршрутизатором PE1 в данные псевдоканала и доставляются им пограничному маршрутизатору PE2, который извлекает эти кадры и отправляет их в сеть *F* в первоначальном виде.

Из определения, данного в RFC 3985, видно, что назначение псевдоканала шире эмуляции Ethernet — это может быть и эмуляции сервисов выделенных каналов технологий PDH или SDH, и эмуляция виртуальных каналов ATM или Frame Relay; однако в любом случае эмуляция такой услуги выполняется через *пакетную сеть*. Тип пакетной сети также не уточняется, так что это может быть и классическая сеть IP (без MPLS), и сеть IP/MPLS, и сеть ATM. Главное в этом обобщенном определении то, что псевдоканал скрывает от пользователей эмулируемого сервиса детали пакетной сети провайдера, соединяя пользовательские пограничные устройства (CE на рис. 21.3) таким образом, как если бы они соединялись с помощью выделенного канала или кабеля.

Для некоторых наиболее важных сочетаний эмулируемого сервиса и типа пакетной сети комитет IETF разработал отдельные спецификации псевдоканалов. Далее мы рассмотрим только один тип псевдоканала, который нужен для предоставления услуг Ethernet операторского класса, а именно — псевдоканал эмуляции Ethernet через сети IP/MPLS, описанный в RFC 4448 (http://www.rfc-editor.org/rfc/rfc4448.txt).

Технически создать LSP второго уровня достаточно просто — для этого маршрутизаторам, соединенным LSP первого уровня, нужно оговорить значение метки второго уровня, которое будет использоваться, чтобы различать LSP второго уровня внутри LSP первого уровня. Этот процесс иллюстрируется рис. 21.4. На нем изображены два пограничных маршрутизатора PE1 и PE2, соединенные псевдоканалом PE57. Однако рисунок оказался немного сложнее, чем можно было предположить — вместо одного пути LSP первого уровня мы видим два таких пути. Это связано с тем, что двухточечные псевдоканалы, которые служат для эмуляции Ethernet, по определению IETF всегда являются двунаправленными[1], а MPLS LSP — это однонаправленный путь. Поэтому для создания двунаправленного псевдоканала требуется два однонаправленных пути второго уровня, вложенных в два однонаправленных пути первого уровня, что и показано на рисунке.

Рассматриваемый в нашем примере псевдоканал в направлении от PE1 к PE2 идентифицируется меткой 57, а туннель, который использует этот канал, — меткой 102. Поэтому при отправке кадра Ethernet, предназначенного для PE2, маршрутизатор PE1 помещает исходный кадр Ethernet в кадр MPLS и адресует этот кадр двумя метками: внешней меткой 102 и внутренней меткой 57. Внешняя метка применяется затем магистральными маршрутизаторами P1, P2 и P3 для того, чтобы доставить кадр пограничному маршрутизатору PE2, при этом в процессе передачи кадра происходит обычная коммутация по меткам (на рисунке показано, что после прохождения P1 внешняя метка получила значение 161). Внутренняя метка 57 требуется только пограничному маршрутизатору PE2, который знает, что эта метка соответствует псевдоканалу PW57, который нужен для связи с некоторой пользовательской сетью.

[1] Форум IETF определил и другие типы псевдоканалов, такие как «точка-многоточка» и «мпоготочка-многоточка». Эти псевдоканалы являются однонаправленными, но для эмуляции Ethernet они не используются.

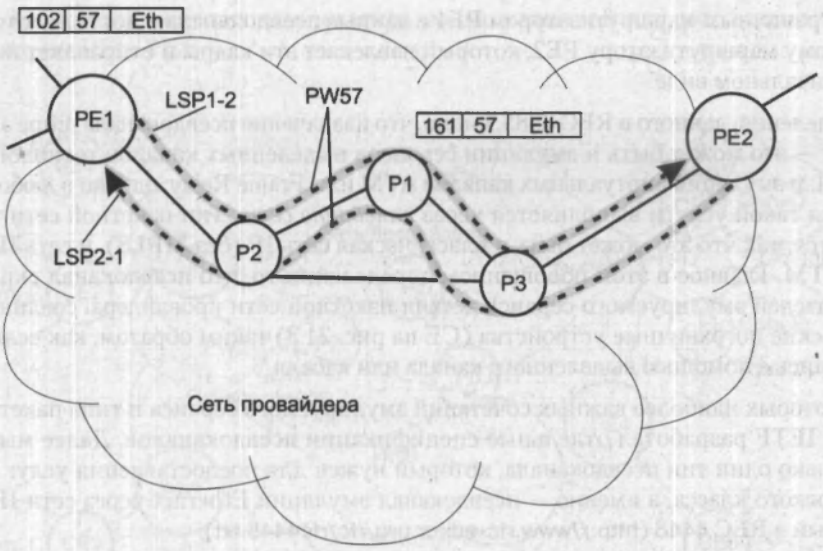


Рис. 21.4. Создание псевдоканала внутри туннелей MPLS

Как мы видим из рассмотренного примера, псевдоканалы работают только внутри сети провайдера, так что для эмуляции сервиса «из конца в конец» нужны еще какие-то элементы и механизмы — и мы скоро их рассмотрим, но сначала давайте обсудим преимущества применения псевдоканалов поверх MPLS. Возникает естественный вопрос: нужны ли они вообще? Нельзя ли просто обойтись LSP первого уровня для передачи трафика Ethernet через сеть провайдера? В принципе, без псевдоканалов обойтись можно, но тогда для каждого нового пользовательского соединения пришлось бы создавать новый туннель (то есть LSP первого уровня), а это не очень масштабируемое решение, так как конфигурирование такого пути обязательно включает конфигурирование всех магистральных маршрутизаторов сети. Поэтому одно из существенных преимуществ псевдоканалов состоит в том, что в сети провайдера нужно сконфигурировать только сравнительно небольшое число туннелей между пограничными маршрутизаторами, а затем использовать каждый из них для прокладки необходимого числа псевдоканалов. Создание нового псевдоканала также требует конфигурирования, но только пары пограничных маршрутизаторов, которые являются конечными точками псевдоканала, а это подразумевает гораздо меньший объем работы.

Можно заметить, что в технике MPLS L3VPN, рассматриваемой в главе 20, также используются пути второго уровня иерархии для соединения пользовательских сайтов в виртуальную частную сеть. Причины применения этого механизма в MPLS L3VPN те же — хорошая масштабируемость.

Другим преимуществом псевдоканалов является их универсальность, то есть возможность их применения не только в сетях MPLS, но и в сетях других типов, например в «чистых» IP-сетях с туннелированием по протоколу L2TP, и не только при эмуляции Ethernet, но и при эмуляции других сервисов, например каналов PDH. Естественно, что при переходе к другой реализации псевдоканалов конкретные команды конфигурирования меняются, но концепция остается, и это помогает администраторам сети освоить новую технологию.

Услуги VPWS

Услуги виртуальных частных каналов (Virtual Private Wire Service, VPWS) исполняют роль «глобального кабеля», соединяя прозрачным образом две локальных пользовательских сети Ethernet через сеть оператора связи. Мы рассмотрим организацию такой услуги с помощью псевдоканалов MPLS на примере (рис. 21.5). При этом мы опишем дополнительные элементы механизма эмуляции услуги Ethernet, которые были опущены при описании назначения псевдоканалов.

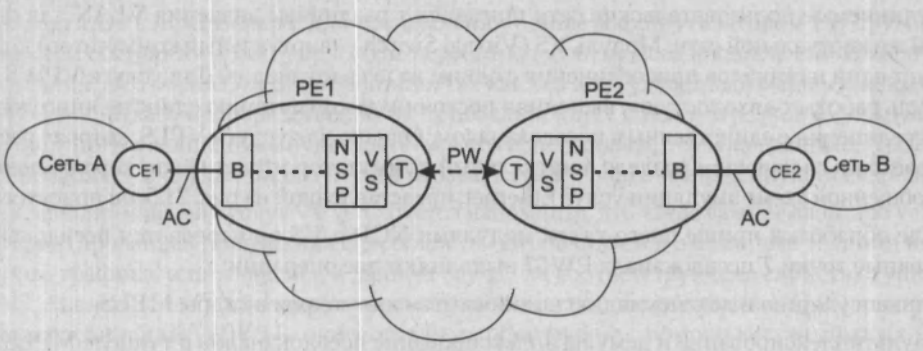


Рис. 21.5. Организация виртуального частного канала Ethernet

Чаще всего пользовательские сети соединяются с пограничным маршрутизатором провайдера через выделенный интерфейс, который для глобальных услуг Ethernet должен быть стандартным интерфейсом Ethernet, например 100Base-FX. В этом случае услуга VPWS заключается в прозрачном соединении этих интерфейсов, когда сеть провайдера передает все кадры, которые поступают на такой интерфейс от сети пользователя. Иногда этот режим VPWS называют коммутацией портов пользователя.

Возможен и другой вариант услуги VPWS, когда сеть провайдера соединяет виртуальные пользовательские сети, то есть по двухточечному соединению передаются не все кадры, поступающие через интерфейс пользователя, а только кадры, принадлежащие определенной сети VLAN. Этот режим работы VPWS можно назвать коммутацией виртуальных локальных сетей, или VLAN-коммутацией.

Для того чтобы обобщить понятие интерфейса с пользователем, форум IETF ввел термин **канала присоединения** (Attachment Circuit, AC). AC поставляет входной поток пользовательских данных для сети провайдера, то есть ту нагрузку, которую нужно коммутировать. Употребляя этот термин, можно сказать, что услуга VPWS всегда соединяет два пользовательских канала присоединения; такое определение справедливо не только для услуг Ethernet, но и для услуг, например, Frame Relay или ATM, в этом случае каналы присоединения являются виртуальными каналами этих технологий.

На рисунке показаны также внутренние функциональные элементы пограничных маршрутизаторов PE1 и PE2, которые эмулируют услуги VPWS вместе с псевдоканалом PW57.

Модуль *B* (от Bridge — мост) работает по стандартному алгоритму IEEE 802.1D. Его роль в схеме эмуляции — выделение кадров Ethernet из общих потоков, поступающих на порты маршрутизатора, для передачи в псевдоканал. Тем самым модуль моста формирует логический интерфейс виртуального коммутатора. Например, если это режим коммутации

портов, то модуль моста конфигурируется так, чтобы все кадры, пришедшие на соответствующий порт от пользователя, направлялись для дальнейшей обработки в псевдоканал. Если же это VLAN-коммутация, то модуль моста выбирает для передачи псевдоканалу только кадры, помеченные определенным значением тега VLAN.

Выбранные модулем моста кадры поступают в псевдоканал не непосредственно, а через два промежуточных модуля — NSP и VS. Модуль NSP (Native Service Processing) обеспечивает предварительную обработку кадров Ethernet. Чаще всего такая обработка связана с изменением или добавлением тега VLAN, что может потребоваться, например, если объединяемые пользовательские сети применяют различные значения VLAN для одной и той же виртуальной сети. Модуль VS (Virtual Switch — виртуальный коммутатор) коммутирует один из каналов присоединения с одним из псевдоканалов. Для услуги VPWS этот модуль работает «вхолостую», выполняя постоянную коммутацию единственного канала присоединения с единственным псевдоканалом. Однако для услуги VPLS, которая рассматривается в следующем разделе, виртуальный коммутатор играет важную роль, поэтому в обобщенной схеме эмуляции услуг Ethernet, представленной на рис. 21.5, он присутствует.

После обработки пришедшего кадра модулями NCP и VS он передается псевдоканалу. Конечные точки *T* псевдоканала PW57 выполняют две операции:

- инкапсуляцию и декапсуляцию пользовательских кадров в кадры MPLS;
- мультиплексирование и демультиплексирование псевдоканалов в туннеле MPLS.

Процедуру инкапсуляции и формат результирующего кадра определяет спецификация RFC 4448. У исходного кадра отбрасываются поля преамбулы и контрольной суммы, после чего он помещается в кадр MPLS с двумя полями меток: внешней (метка туннеля) и внутренней (метка псевдоканала), как это показано на рис. 21.6). На рисунке не показаны поля заголовка кадра MPLS, относящиеся к конкретной канальной технологии, которая используется на внутренних интерфейсах пограничных маршрутизаторов — как вы помните, кадры MPLS могут иметь обрамление Ethernet, PPP, ATM или Frame Relay (в случае Ethernet это обрамление не имеет отношения к пользовательскому кадру Ethernet, инкапсулированному в кадр MPLS).

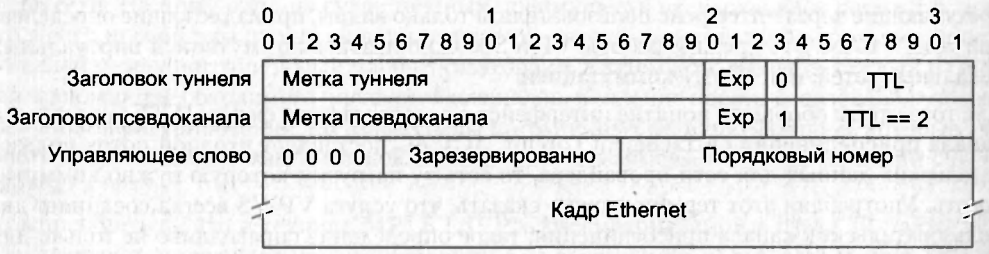


Рис. 21.6. Формат инкапсуляции Ethernet поверх MPLS (RFC 4448)

В то время как первые два слова в заголовке, представленном на рисунке, являются стандартными заголовками MPLS, третье слово, называемое управляющим (control word), впервые появилось в стандарте RFC 4448. Это слово, которое является опциональным, предназначено для упорядочивания кадров, передаваемых по псевдоканалу — для этого каждому кадру маршрутизатором-отправителем присваивается порядковый номер, который помещается в управляющее слово. Потребность в контрольном слове возникает тогда,

когда внутри сети провайдера происходит распараллеливание трафика туннеля, и кадры могут выходить из туннеля не в том порядке, в котором были посланы.

Конфигурирование псевдоканалов, то есть согласование внутренних меток, используемых для идентификации и мультиплексирования псевдоканалов внутри туннеля, может быть автоматизировано. Для этого сегодня применяют протокол LDP или BGP. Обратите внимание, что речь идет о прокладке псевдоканала, а не самого туннеля, эти два процесса независимы, так что туннель может быть проложен, например, с помощью протокола RSVP TE, а псевдоканалы в нем — с помощью протокола LDP.

Протокол LDP служит также для уведомления одним маршрутизатором PE другого об изменении состояния «работоспособен-неработоспособен» псевдоканала или канала присоединения. Это очень полезное свойство, так как без него удаленный маршрутизатор PE не узнает об отказе непосредственно не присоединенных к нему отрезков эмулируемого транспортного соединения и будет пытаться его использовать, посылая данные. Протокол LDP позволяет в случае такого отказа отозвать метку, ранее назначенную псевдоканалу.

В завершение описания услуг VPWS хочется напомнить, что такое важное свойство услуги, как гарантированная пропускная способность, обеспечивается с помощью техники инжиниринга трафика, опирающейся в данном случае на соответствующие свойства туннелей MPLS. Аналогично обстоит дело с параметрами качества обслуживания (QoS) для виртуальных соединений VPWS — они могут быть обеспечены с помощью стандартных механизмов QoS, таких как, например, приоритетное обслуживание, профилирование трафика, контроль доступа и резервирование ресурсов. И в этом случае MPLS является хорошим базисом, так как детерминированность туннелей MPLS делает контроль доступа намного более определенной процедурой, чем в случае IP-сетей с их распределенным (и вносящим неопределенность) механизмом выбора маршрутов.

Услуги VPLS

Услуги виртуальной частной локальной сети (Virtual Private LAN Service, VPLS) описаны в спецификациях RFC 4761 (http://www.rfc-editor.org/rfc/rfc4761.txt) и RFC 4762 (http://www.rfc-editor.org/rfc/rfc4762.txt).

Услуги VPLS соответствуют определению услуг E-LAN MEF, причем как варианту с учетом идентификаторов VLAN пользователей, так и варианту без их учета.

Так же как и в случае VPWS, сервис VPLS организован на базе псевдоканалов. Отличие заключается в том, что для каждого экземпляра VPLS используется собственный набор псевдоканалов. При этом каждый такой набор имеет полносвязную топологию, то есть все пограничные маршрутизаторы PE, участвующие в работе какого-то экземпляра VPLS, связаны друг с другом.

На рис. 21.7 показан пример сети провайдера, эмулирующей два сервиса VPLS. Пользовательские сети C1, C5 и C8 относятся к «серому» сервису VPLS, а сети C2, C3, C4, C6 и C7 — к «белому». Соответственно, набор псевдоканалов PW-B1, PW-B2 и PW-B3 объединяет пограничные маршрутизаторы, к которым подключены сети «серого» сервиса VPLS, а набор псевдоканалов PW-W1, PW-W2 и PW-W3 — маршрутизаторы, к которым подключены сети «белого» сервиса VPLS (в нашем примере это одни и те же пограничные маршрутизаторы PE1, PE2 и PE3, но если бы, например, сети C4 не существовало, то псевдоканалы PW-W2 и PW-W3 были бы не нужны).

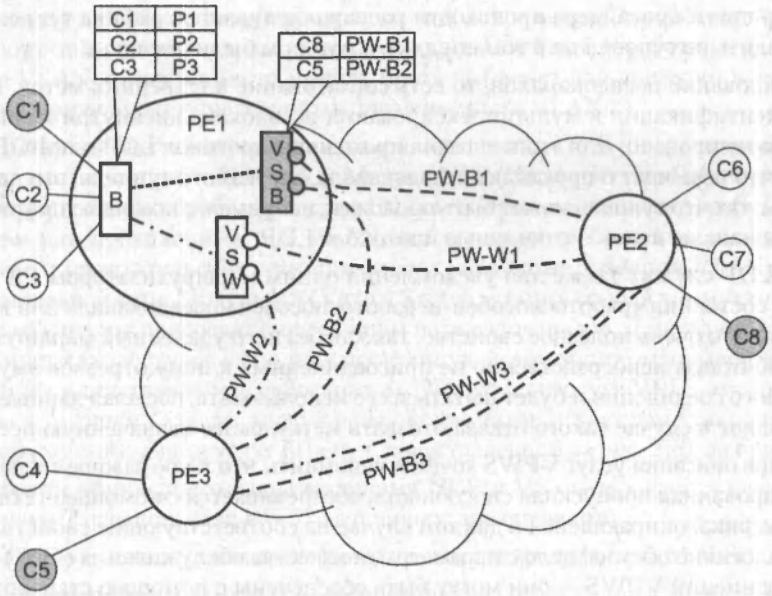


Рис. 21.7. Организация услуги VPLS

Внутренняя организация пограничного маршрутизатора при оказании услуги VPLS показана на примере маршрутизатора PE1. Мы видим, что для поддержки каждого экземпляра сервиса VPLS пограничному маршрутизатору требуется отдельный виртуальный коммутатор, в данном случае это модули VPB и VPW (модули NSP не показаны, чтобы не загромождать рисунок, но они в PE1 входят, по одному на каждый экземпляр VPLS).

Как и в случае VPWS, модуль *B* выполняет стандартные функции моста и при этом формирует логический интерфейс с каждым из виртуальных коммутаторов. Этот интерфейс может также формироваться на основе коммутации либо пользовательских портов, когда весь трафик от определенного порта (или нескольких портов) передается на логический интерфейс, либо сетей VLAN, когда выбираются кадры одной или нескольких пользовательских сетей VLAN от одного или нескольких портов.

Однако если в случае VPWS виртуальный коммутатор выполнял простую работу по передаче кадров от логического интерфейса, то для VPLS этот модуль функционирует по алгоритму стандартного коммутатора (моста). Для этого виртуальный коммутатор изучает MAC-адреса и строит свою таблицу продвижения, как и обычный коммутатор. На рисунке показан упрощенный вид таблицы продвижения PE1, состоящей из двух записей: одна запись связывает адрес M8 сети C8 с псевдоканалом PW-B1, другая — адрес M5 сети C5 с псевдоканалом PW-B2. Пользуясь такой таблицей, виртуальный коммутатор не затапливает сеть, получая кадры с адресами M5 или M8, а направляет их в псевдоканал, ведущий к пограничному коммутатору, к которому подключена сеть с узлом назначения. Кадры с широковещательным адресом или адресом, отсутствующим в таблице продвижения, поступают на все его псевдоканалы, в данном случае — на PW-B1 и PW-W1.

Единственной особенностью виртуального коммутатора является то, что он не изучает адреса отправления кадров, приходящих с логического интерфейса. Это не требуется, потому что для интерфейсов, представленных псевдоканалами, виртуальный коммутатор

работает по правилу расщепления горизонта (split horizon) — он никогда не передает на псевдоканалы кадры, полученные от какого бы то ни было псевдоканала. Тем самым предотвращается образование петель между виртуальными коммутаторами, а доставку кадров по назначению гарантирует полносвязная топология. То есть любой кадр, полученный виртуальным коммутатором по псевдоканалу, всегда передается на логический интерфейс, соответствующий тому сервису VPLS, к которому относится псевдоканал.

Модуль моста *B* изучает только адреса, приходящие с пользовательских интерфейсов. Они служат ему для выбора нужного интерфейса в том случае, когда несколько пользовательских сетей относятся к одному сервису VPLS.

Конфигурирование PE может оказаться трудоемким занятием, так как в случае N пограничных коммутаторов нужно создать $N(N-1)/2$ псевдоканалов. Кроме того, добавление любого нового устройства PE требует переконфигурирования всех остальных коммутаторов. Для автоматизации этих процедур можно использовать вариант организации VPLS, описанный в RFC 4761, так как он предусматривает применение для этой цели протокола BGP. Вариант VPLS, описанный в RFC 4762, подразумевает распределение меток второго уровня иерархии с помощью протокола LDP, автоматизацию процедур конфигурирования он не поддерживает.

Ethernet поверх Ethernet

Области улучшений Ethernet

Рассмотрим более подробно те новые свойства, которые необходимо добавить к классическому варианту Ethernet, чтобы превратить его в транспортную технологию операторского класса (Carrier Ethernet Transport, CET), способную работать в сети провайдера в качестве основного транспортного механизма.

Разделение адресных пространств пользователей и провайдера

Адресное пространство сети современной коммутируемой сети Ethernet состоит из двух частей: значений MAC-адресов конечных узлов и значений меток локальных виртуальных сетей (VLAN), на которые логически разделена сеть. Коммутаторы Ethernet при принятии решения при продвижении кадра учитывают оба адресных параметра.

Если сеть провайдера будет составлять с сетями пользователей единое целое на уровне Ethernet, то такая сеть окажется практически неработоспособной, так как все коммутаторы провайдера должны будут в своих таблицах продвижения содержать MAC-адреса всех конечных узлов всех пользователей, а также поддерживать принятое каждым пользователем разбиение сети на локальные виртуальные сети. Помимо очевидной проблемы с количества MAC-адресов (для крупного провайдера это значение может доходить до нескольких миллионов), есть еще проблема с их уникальностью — хотя система назначения адресов и призвана предотвратить дублирование «аппаратных» MAC-адресов, существуют еще и программируемые адреса, да и ошибки в прошивании аппаратных адресов тоже случаются.

Использование пользовательских меток VLAN в сети провайдера также приводит к проблемам. Во-первых, пользователям нужно договариваться о согласованном применении

значений VLAN, чтобы они были уникальными для каждого пользователя, так как только тогда сеть провайдера сможет доставлять кадры нужным пользовательским сетям. Представить, как реализовать такую процедуру практически, очень непросто, ведь каждый новый пользователь приходит со своими значениями VLAN, и если заставлять его их переназначать, то можно потерять пользователя. Во-вторых, стандарт VLAN изначально не был рассчитан на глобальное применение и поэтому в нем предусмотрено только 4092 значения метки, что крайне мало для крупного провайдера.

Если посмотреть, как решаются эти проблемы в сетях провайдеров, построенных на других принципах, то мы увидим, что при использовании провайдером технологии IP MAC-адреса пользователей вообще не проникают в маршрутизаторы провайдера[1], а IP-адреса пользователей представлены в таблицах маршрутизаторов в агрегированном виде — прием, для плоских MAC-адресов недоступный. В сетях, реализующих рассмотренную ранее технологию EoMPLS, MAC-адреса и метки VLAN пользователей применяются только в пограничных маршрутизаторах провайдера, а в магистральных маршрутизаторах они не работают — там их заменяют два уровня меток MPLS.

Маршрутизация, инжиниринг трафика и отказоустойчивость

Операторы связи привыкли к ситуации полного контроля над путями следования трафика в своих сетях, что обеспечивает, например, технология SDH. В IP-сетях степень контроля оператора над маршрутами трафика очень низкая, и одной из причин популярности технологии MPLS служит то, что она привнесла в IP-сети детерминированность маршрутов. Другой желательной для операторов характеристикой сети является отказоустойчивость маршрутов, то есть возможность быстрого перехода на новый маршрут при отказах узлов или линий связи сети. Технология SDH всегда была в этом плане эталоном, так как обеспечивает переход с основного на заранее проложенный резервный путь за десятки миллисекунд. MPLS также обладает подобным свойством.

В сетях Ethernet маршрутизация трафика и отказоустойчивость обеспечиваются протоколом покрывающего дерева (STP). Этот протокол дает администратору сети очень ограниченный контроль над выбором маршрута (это справедливо и для новых вариантов STP, таких как RSTP и MSTP). Кроме того, покрывающее дерево является общим для всех потоков независимо от их адреса назначения. Ввиду этих особенностей протокол STP/RTP является очень плохим решением в отношении инжиниринга трафика. Отказоустойчивость маршрутов также обеспечивается STP, и хотя новая версия RTP значительно сократила время переключения на новый маршрут (с нескольких десятков секунд до одной-двух), до миллисекундного диапазона SDH ей очень далеко. Все это требует нового подхода к маршрутизации потоков в сетях CET, и IEEE работает над этой проблемой.

Функции эксплуатации, администрирования и обслуживания

Функции эксплуатации, администрирования и обслуживания (Operation, Administration, Maintenance, OAM) всегда были слабым звеном Ethernet, и это одна из главных причин,

[1] Если быть предельно педантичным, нужно сделать оговорку: за исключением MAC-адресов пограничных интерфейсов пользовательских маршрутизаторов, которые попадают в ARP-таблицы интерфейсов пограничных машрутизаторов провайдера в случае, если это интерфейсы Ethernet.

по которой операторы связи не хотят применять эту технологию в своих сетях. Новые стандарты, предлагаемые IEEE и ITU-T, призваны исправить эту ситуацию, вводя средства, с помощью которых можно выполнять мониторинг достижимости узлов, локализовывать неисправные сегменты сети и измерять уровень задержек и потерь кадров между узлами сети.

Первая группа функций направлена на решение проблемы использования Ethernet для оказания услуги виртуальных частных сетей, а две остальные — на придание Ethernet функциональности, необходимой для применения Ethernet в качестве внутренней транспортной технологии оператора связи.

Функции эксплуатации, администрирования и обслуживания в Ethernet

К настоящему времени разработано несколько стандартов Ethernet, относящихся к функциям эксплуатации, администрирования и обслуживания:

- IEEE 802.1ag. Connectivity Fault Management (CFM). Стандарт описывает протокол мониторинга состояния соединений, в какой-то степени это аналог протокола BFD, рассмотренного в главе 20.
- ITU-T Y.1731. Стандарт комитета ITU-T воспроизводит функции стандарта IEEE 802.1ag и расширяет их за счет группы функций мониторинга параметров QoS.
- IEEE 802.3ah. Стандарт тестирования физического соединения Ethernet.
- MEF E-LMI. Интерфейс локального управления Ethernet.

Протокол CFM

Протокол CFM обеспечивает мониторинг логических соединений различного типа, например это может быть соединение определенной сети VLAN или же соединение EoMPLS услуги VPWS. Протокол CFM может выполнять мониторинг как непосредственно соединенных узлов, так и узлов, соединение между которыми проходит через несколько сетей. Кроме того, CFM может использоваться для соединений полносвязной топологии, характерных для услуг типа E-LAN.

Мониторинг выполняется между так называемыми конечными точками обслуживания (Maintenance End Point, MEP), представляющих собой конечные точки соединения, состояние которого нужно наблюдать.

Каждая из точек MEP периодически посылает сообщения проверки непрерывности соединения (Continuity Check Message, CCM), оформленные как кадры сервиса, соединение которого тестируется. Например, если тестируется соединение по VLAN 5, то сообщения CCM оформляются как кадры Ethernet с идентификатором VLAN, равным 5.

Устройства, которые не имеют точек MEP, передают такие сообщения транзитом. В том случае, когда некоторая точка MEP не принимает сообщений CCM от другой точки MEP в течение заданного тайм-аута, соединение считается неработоспособным.

В промежуточных устройствах, через которые проходит соединение, можно сконфигурировать промежуточные точки обслуживания (Maintenance Intermediate Point, MIP). Эти точки помогают отслеживать проблемы, возникающие на промежуточных устройствах.

На рис. 21.8 показан случай мониторинга состояния соединения через сеть VLAN 5. Для этого служат три точки MEP, одна из которых располагается в сети провайдера, а две

другие — в пограничном оборудовании пользователя. Для того чтобы осуществлять мониторинг соединения полносвязной топологии, которое представляет собой VLAN 5, сообщения CCM посылаются с групповым адресом Ethernet. Для мониторинга двухточечных соединений могут использоваться как индивидуальные, так и групповые адреса.

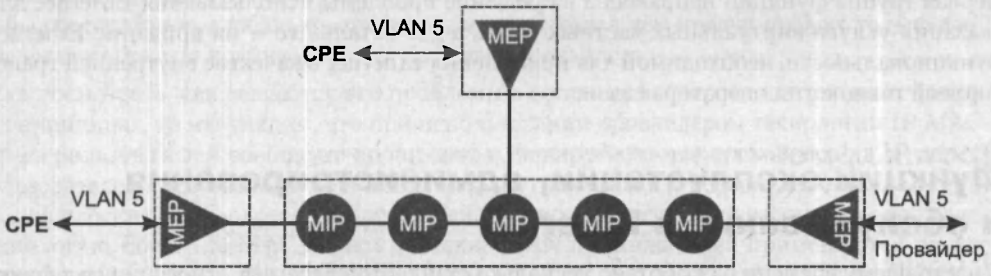


Рис. 21.8. Мониторинг состояния VLAN с помощью протокола CFM

Весьма важной является способность протокола CFM работать в многодоменной среде, когда соединение проходит через несколько сетей, принадлежащих различным административным доменам. Такая ситуация обычно возникает, если соединение является соединением виртуальной частной сети, организуемой одним или несколькими провайдерами (например, когда поставщик услуги VPN пользуется для организации своей сети услугами выделенных каналов оператора связи). Каждый из администраторов доменов нуждается в мониторинге соединения, но только в пределах своей сети.

Для поддержки многодоменного сценария для каждого домена конфигурируется отдельный домен обслуживания, при этом домены обслуживания образуют иерархию доменов, то есть каждый домен работает на своем индивидуальном уровне. В каждом домене создаются точки обслуживания MEP и MIP, но точки каждого домена работают только с сообщениями CCM своего уровня, а сообщения более высоких уровней просто прозрачно передают.

Эту идею иллюстрирует рис. 21.9. Здесь показана сеть, состоящая из трех доменов: домена пользователя, домена поставщика услуги виртуальной частной сети и домена оператора связи, через который работает сеть поставщика услуги.

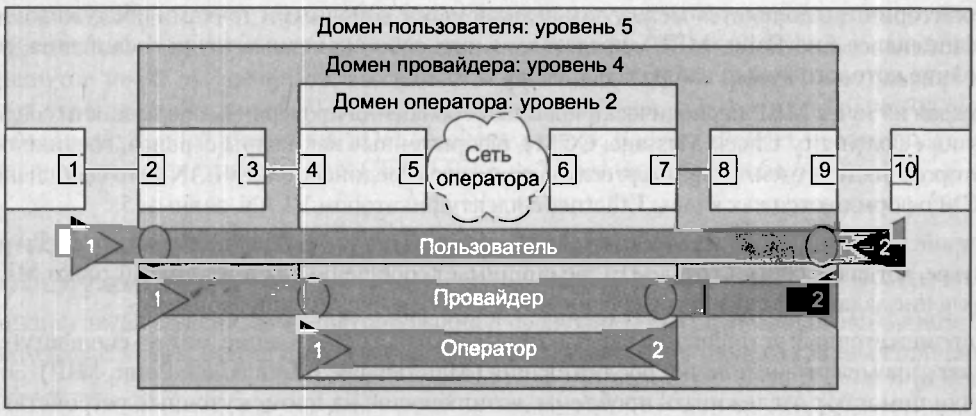


Рис. 21.9. Многодоменное применение протокола CFM

Домену пользователя присвоен уровень 5, домену провайдера — уровень 4, домену оператора связи — уровень 2 (уровнем по умолчанию в протоколе CFM является уровень 3, он в этом примере отсутствует). Точки обслуживания в сети оператора связи работают с сообщениями CCM уровня 2, а сообщения точек обслуживания сети пользователя уровня 5 и сети поставщика услуги уровня 4 они передают прозрачно.

В результате оператор связи получает информацию о состоянии соединения в пределах своей сети, провайдер — в пределах своей, а пользователь соединения — «из конца в конец».

Протокол мониторинга качества соединений Y.1731

Стандарт Y.1731, разработанный ITU-T, добавляет к стандарту CFM возможность измерять между точками обслуживания сети некоторые дополнительные параметры.

- *Односторонняя задержка кадра.* Для измерения этой задержки точки обслуживания сети MEP генерируют сообщения измерения задержки и ответа на измерение задержки. В этих сообщениях переносятся временные отметки, позволяющие измерить задержку.
- *Вариация задержки.* Эта задержка измеряется на основе тех же сообщений, что и односторонняя задержка.
- *Потери кадров.* Для измерения этой величины служат сообщения измерения потерь и ответа на измерение потерь. Счетчики сообщений двух точек обслуживания сравниваются и на основе этого сравнения рассчитываются потери кадров в каждом из направлений.

Стандарт тестирования физического соединения Ethernet

Стандарт тестирования физического соединения Ethernet предназначен для обнаружения ошибок соединения между двумя непосредственно физически связанными интерфейсами Ethernet. Он поддерживает такие функции, как удаленное обнаружение неисправностей и удаленный контроль обратной связи.

Последняя функция является наиболее интересной для специалистов, занимающихся эксплуатацией сетей Ethernet, так как она позволяет удаленно (через сеть) выдать запрос некоторому интерфейсу Ethernet на переход в режим обратной связи. В этом режиме все кадры, посылаемые на этот интерфейс соседом по линии связи, возвращаются им обратно. Полученные кадры затем можно проанализировать, чтобы установить качество физической линии.

Необходимо отметить, что процедура тестирования линии в режиме обратной связи нарушает нормальную работу соединения, поэтому тестирование нужно проводить в специальное время, отведенное под обслуживание сети.

Интерфейс локального управления Ethernet

Стандарт E-LMI позволяет пограничному пользовательскому устройству, то есть устройству типа CE, запрашивать информацию о состоянии и параметрах услуги, предоставляемой сетью провайдера по данному интерфейсу. Например, пограничный коммутатор Ethernet, расположенный в сети пользователя, может запросить у пограничного коммутатора провайдера (то есть устройства PE) информацию о состоянии услуги E-LINE или

E-LAN, предоставляемой по данному интерфейсу. Кроме того, согласно стандарту E-LMI, по запросу можно получить такую информацию об услуге, как отображение идентификатора VLAN пользователя на соединение EVC, характеризующее номер виртуальной частной сети, или же величина пропускной способности, гарантированной для данного соединения EVC.

Мосты провайдера

Стандарт IEEE 802.1ad «Мосты провайдера» (Provider Bridge, PB) был первым стандартом, который решал проблему изоляции адресного пространства сети провайдера от адресного пространства его пользователей. Этот стандарт был принят IEEE в 2005 году, и сегодня он реализован в коммутаторах Ethernet многих производителей.

Нужно сказать, что проблема изоляции адресных пространств решается в этом стандарте только частично, так как MAC-адреса пользователей по-прежнему присутствуют в коммутаторах сети провайдера, разделяются только пространства идентификаторов VLAN.

Стандарт PB вводит двухуровневую иерархию идентификаторов VLAN (рис. 21.10). На внешнем (верхнем) уровне располагается идентификатор VLAN провайдера, называемый S-VID (от Service VLAN ID – идентификатор сервиса VLAN), а на нижнем (внутреннем) уровне – идентификатор VLAN пользователя, называемый C-VID (от Customer VLAN ID – идентификатор VLAN потребителя).

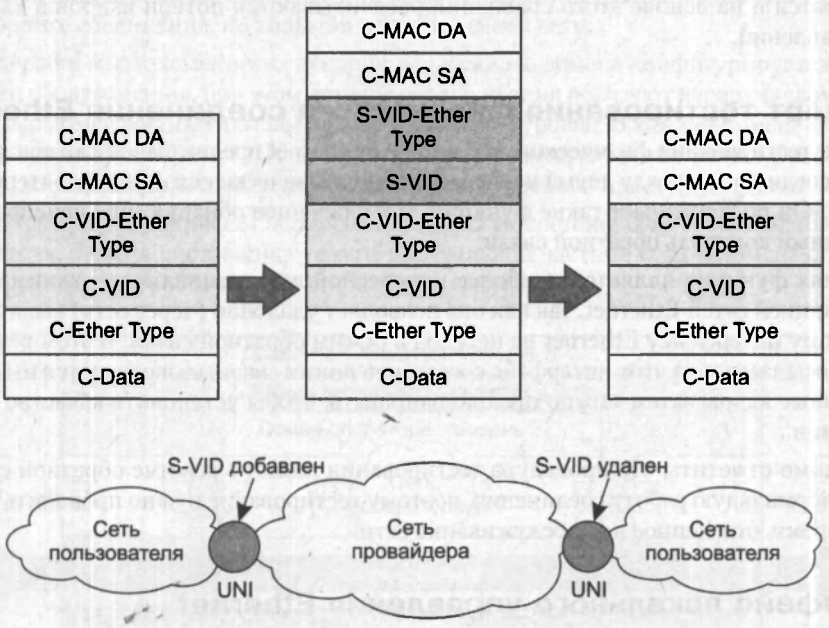


Рис. 21.10. Инкапсуляция идентификаторов VLAN

Идентификатор S-VID помещается в пользовательский кадр пограничным коммутатором провайдера, он просто проталкивает C-VID в стек и добавляет новый идентификатор

S-VID, который потребуется коммутаторам сети провайдера для разделения трафика на виртуальные локальные сети внутри сети провайдера. Так как S-VID представляет собой новое поле кадра Ethernet, то ему предшествует новое поле типа EtherType, которое на рис. 21.10 обозначено как S-VID-EtherType (в отличие от оригинального поля C-VID-EtherType). Для отличия S-VID от C-VID стандарт 802.1ad вводит новое значение EtherType 0x88a8 для типа данных S-VID (напомним, что для C-VID используется значение EtherType 0x8100). Этот способ инкапсуляции часто неформально называют инкапсуляцией Q-in-Q по названию стандарта 802.1Q, описывающего технику VLAN.

После того как пограничный коммутатор сети провайдера выполняет инкапсуляцию, кадр обрабатывается магистральными коммутаторами провайдера как обычный кадр, поэтому эти коммутаторы не обязаны поддерживать стандарт 802.1ad (за исключением поддержки нового значения EtherType 0x88a8, но его использование не является обязательным, и многие производители коммутаторов Ethernet допускают конфигурирование этого параметра и применение стандартного значения 0x8100 и для S-VID).

Когда кадр прибывает на выходной пограничный коммутатор провайдера, над ним выполняется обратная операция — идентификатор S-VID удаляется. После этого кадр отправляется в сеть пользователя в исходном виде, имея в своем заголовке только идентификатор C-VID.

Внутренние сети VLAN провайдера, соответствующие значениям идентификаторов S-VID, обычно служат для конструирования услуг типа E-LAN. При этом провайдеру нет необходимости согласовывать логическую структуру своей сети с пользователями.

На рис. 21.11 показана сеть провайдера, которая предоставляет потребителям две услуги типа E-LAN. Сайты C1, C3 и C5 относятся к сервису E-LAN с идентификатором S-VID 156, а сайты C2, C4 и C6 — к сервису E-LAN с идентификатором S-VID 505.

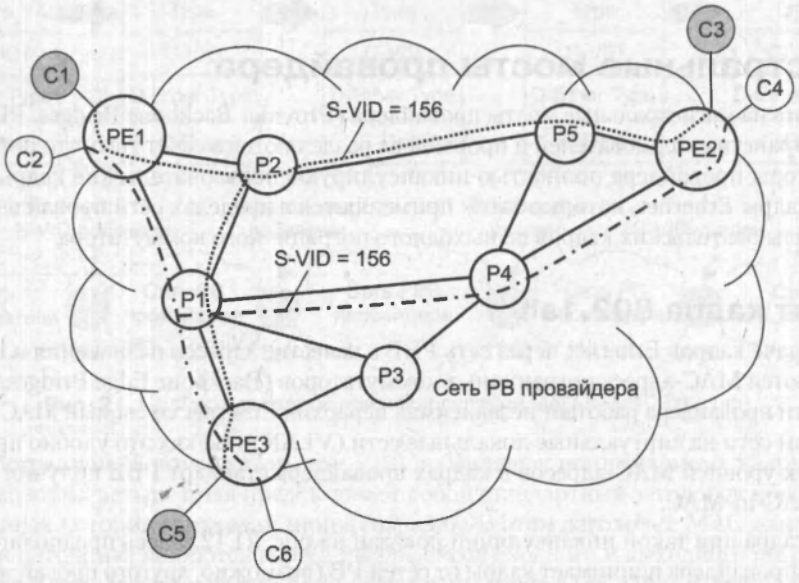


Рис. 21.11. Сеть стандарта PB, предоставляющая две услуги типа E-LAN

Конфигурирование услуг E-LAN 156 и 505 выполнено без учета значений пользовательских идентификаторов VLAN на основании подключения сайта пользователя к некоторому физическому интерфейсу коммутатора провайдера. Так, например, весь пользовательский трафик, поступающий от сайта C1, классифицируется пограничным коммутатором PE1 как принадлежащий к виртуальной частной сети с S-VID 156.

В то же время стандарт PB позволяет провайдеру предоставлять услуги и с учетом значений пользовательских идентификаторов VLAN. Например, если внутри сайта C1 выполнена логическая структуризация и существуют две пользовательские сети VLAN, трафик которых нельзя смешивать, провайдер может организовать для этого две сети S-VLAN и отображать на них поступающие кадры в зависимости от значений C-VID.

При своей очевидной полезности стандарт PB имеет несколько недостатков.

- Коммутаторы сети провайдера, как пограничные, так и магистральные, должны изучать MAC-адреса узлов сетей пользователей. Это не является масштабируемым решением.
- Максимальное количество услуг, предоставляемых провайдером, ограничено числом 4096 (так как поле S-VID имеет стандартный размер в 12 бит).
- Инжиниринг трафика ограничен возможностями протокола покрывающего дерева RSTP/MSTP.
- Для разграничения деревьев STP, создаваемых в сетях провайдера и пользователей, в стандарте 802.1ad пришлось ввести новый групповой адрес для коммутаторов провайдера. Это обстоятельство не позволяет задействовать в качестве магистральных коммутаторов провайдера те коммутаторы, которые не поддерживают стандарт 802.1ad.

Некоторые из этих недостатков были устранены в стандарте IEEE 802.1ah, который был принят летом 2008 года.

Магистральные мосты провайдера

В стандарте на магистральные мосты провайдера (Provider Backbone Bridges, PBB) адресные пространства пользователей и провайдера разделяются за счет того, что пограничные коммутаторы провайдера полностью инкапсулируют пользовательские кадры Ethernet в новые кадры Ethernet, которые затем применяются в пределах сети провайдера для доставки пользовательских кадров до выходного пограничного коммутатора.

Формат кадра 802.1ah

При передаче кадров Ethernet через сеть PBB в качестве адресов назначения и источника используются MAC-адреса пограничных коммутаторов (Backbone Edge Bridges, BEB). По сути, в сети провайдера работает независимая иерархия Ethernet со своими MAC-адресами и делением сети на виртуальные локальные сети (VLAN) так, как это удобно провайдеру. Из-за двух уровней MAC-адресов в кадрах провайдера стандарт PBB получил также название MAC-in-MAC.

Формат кадра при такой инкапсуляции показан на рис. 21.12. Здесь предполагается, что сеть PBB провайдера принимает кадры от сетей PB (возможно, другого провайдера), которые, в свою очередь, соединены с сетями пользователя. В этом случае интерфейсы между сетью PBB и сетями PB носят название NNI (Network to Network Interface — интерфейс

«сеть-сеть»), а в поступающих на пограничные коммутаторы сети PBB кадрах имеется идентификатор S-VID, добавленный входным пограничным коммутатором сети PB (и не удаленный выходным пограничным коммутатором сети PB, так как такое удаление выполняется для интерфейсов UNI, но не для интерфейсов NNI). Наличие идентификатора S-VID во входных кадрах не является необходимым условием работы сети PBB, это только возможный вариант; если сеть PBB непосредственно соединяет сети пользователей, то входящие кадры поля S-VID не имеют.

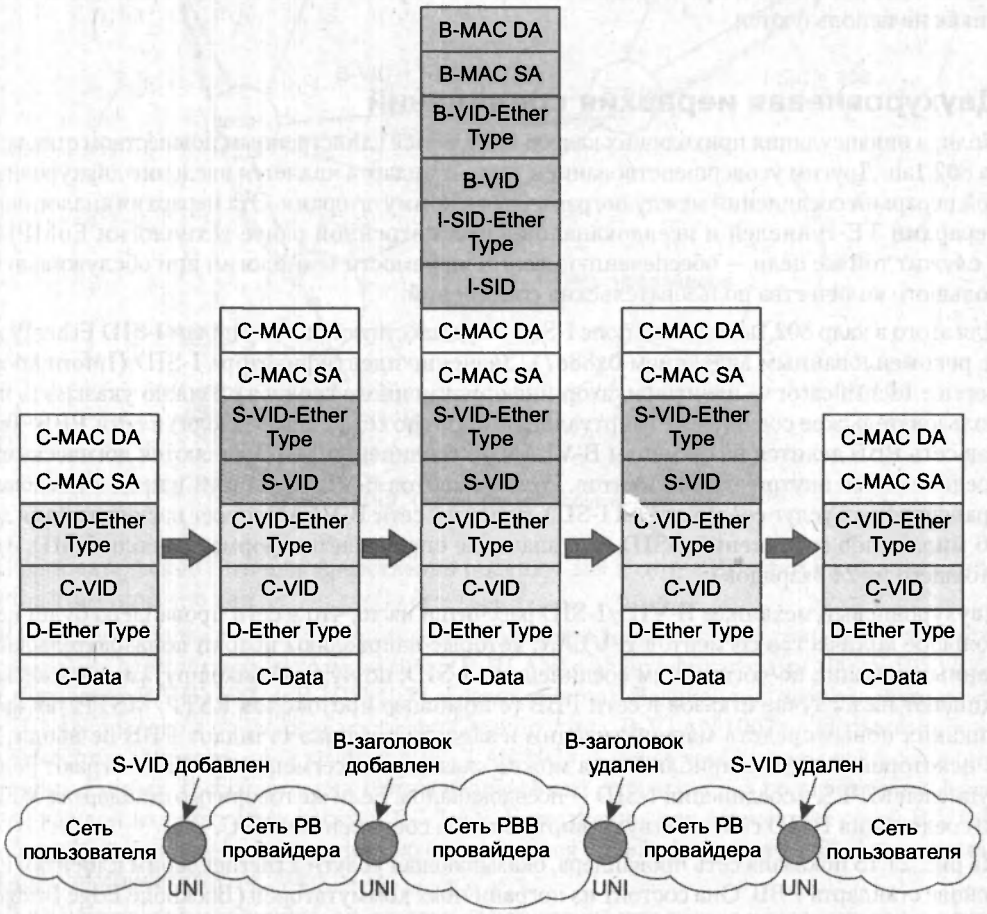


Рис. 21.12. Формат кадров при инкапсуляции MAC-in-MAC 802.1ah

Входной пограничный коммутатор сети PBB добавляет к принимаемому кадру 6 новых полей, из которых четыре поля представляют собой стандартный заголовок нового кадра, в поле данных которого упакован принятый кадр. В этом заголовке MAC-адресами назначения и источника являются адреса интерфейсов входного и выходного пограничных коммутаторов сети, которые на рис. 21.12 обозначены как B-MAC DA и B-MAC SA соответственно (буква «B» в этих обозначениях появилась от слова «backbone» — магистральный). Эти адреса используются в пределах сети PBB вместе с идентификатором виртуальной

локальной сети B-VID для передачи кадров в соответствии со стандартной логикой локальной сети, разделенной на сегменты VLAN, и при этом совершенно независимо от адресной информации сетей пользователя. В качестве значения EtherType для B-VID стандарт 802.1ah рекомендует применять значение 0x88a8, как и для S-VID в стандарте 802.1ad, но допустимы и другие значения, например стандартное для C-VID значение 0x8100 (как и для сетей PB эта возможность зависит от решения производителя оборудования).

Пользовательские MAC-адреса, а также идентификаторы S-VID и C-VID находятся в поле данных нового кадра и при передаче между магистральными коммутаторами сети PBB никак не используются.

Двухуровневая иерархия соединений

Полная инкапсуляция приходящих кадров не является единственным новшеством стандарта 802.1ah. Другим усовершенствованием этого стандарта является введение двухуровневой иерархии соединений между пограничными коммутаторами. Эта иерархия аналогична иерархии TE-туннелей и псевдоканалов в рассмотренной ранее технологии EoMPLS и служит той же цели — обеспечению масштабируемости технологии при обслуживании большого количества пользовательских соединений.

Для этого в кадр 802.1ah введено поле I-SID с предшествующим ему полем I-SID EtherType (с рекомендованным значением 0x88e7). Значение идентификатора I-SID (Information Service Identificator — идентификатор информационного сервиса) должно указывать на пользовательское соединение (виртуальную частную сеть пользователя) в сети PBB. Так как сеть PBB делится на сегменты B-VLAN, то соединения I-SID являются логическими соединениями внутри этих сегментов. Роль сегментов B-VLAN состоит в предоставлении транспортных услуг соединениям I-SID, в каждой сети B-VLAN может насчитываться до 16 миллионов соединений I-SID (это значение определяется форматом поля I-SID, состоящего из 24 разрядов).

Двухуровневый механизм B-VID/I-SID рассчитан на то, что в сети провайдера будет небольшое количество сегментов B-VLAN, которые направляют потоки пользовательских данных, идущих по логическим соединениям I-SID, по нужным маршрутам, а также защищают их в случае отказов в сети PBB (с помощью протоколов RSTP/MSTP, так как никаких новых средств маршрутизации и защиты трафика стандарт PBB не вводит). С некоторой степенью приближения можно сказать, что сегменты B-VLAN играют роль туннелей MPLS, а соединения I-SID — псевдоканалов. Если же говорить о стандартах MEF, то соединения I-SID соответствуют виртуальным соединениям EVC.

На рис. 21.13 показана сеть провайдера, оказывающая услуги Ethernet своим клиентам на основе стандарта PBB. Она состоит из пограничных коммутаторов (Backbone Edge Bridge, BEB) и магистральных коммутаторов (Backbone Core Bridge, BCB).

Провайдер в этом примере предоставляет услуги трех частных виртуальных сетей:

- E-LINE1 — передает голосовой трафик между сетями C1 и C3 (двухточечная топология);
- E-LINE2 — передает голосовой трафик между сетями C2 и C4 (двухточечная топология);
- E-LAN1 — передает эластичный трафик данных между сетями C2, C4 и C6 (полносвязная топология).

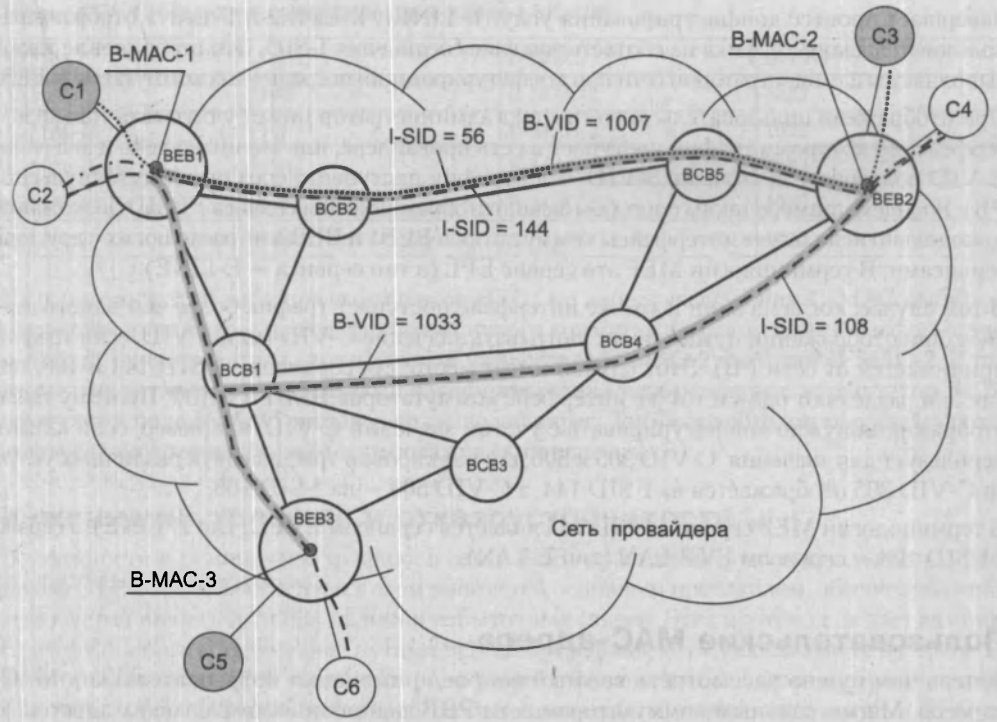


Рис. 21.13. Организация услуг в сети PBB

Пользовательские сети непосредственно подключены к сети PBB, промежуточных сетей PB в этом примере нет.

На верхнем уровне структуризации сети провайдера в ней сконфигурированы две *магистральные виртуальные локальные сети* (B-VLAN) с идентификаторами 1007 и 1033 (обозначены как B-VID 1007 и B-VID 1033 соответственно). В нашем примере различные сети B-VLAN призваны поддерживать трафик разного типа: B-VLAN 1007 поддерживает более требовательный голосовой трафик, а B-VLAN 1033 — менее требовательный эластичный трафик данных. В соответствии с этим назначением созданы и два покрывающих дерева для каждой из виртуальных сетей B-VLAN. Естественно, что назначение сетей B-VLAN может быть и иным — оно полностью определяется оператором сети PBB в соответствии с его потребностями.

На уровне пользовательских услуг в сети организовано три пользовательских соединения, помеченные как I-SID 56, 144 и 108. Эти соединения предназначены для реализации услуг E-LINE1, E-LINE2 и E-LAN1 соответственно.

Соединения I-SID 56 и 144 отображаются пограничными коммутаторами BEB1 и BEB2 на B-VLAN 1007, так как эти соединения переносят пользовательский голосовой трафик, а данная сеть B-VLAN была создана для этого типа трафика. В то же время соединение I-SID 108 отображается пограничными коммутаторами BEB1, BEB2 и BEB3 на B-VLAN 1033, так как сервис 108 переносит эластичный пользовательский трафик данных. Задает эти отображения администратор при конфигурировании пограничных коммутаторов.

Завершает процесс конфигурирования услуг E-LINE1, E-LINE2 и E-LAN1 отображение пользовательского трафика на соответствующие соединения I-SID. Это отображение также выполняется администратором сети при конфигурировании пограничных коммутаторов BEB.

При отображени пользовательского трафика администратор может учитывать только интерфейс, по которому трафик поступает в сеть провайдера, или же интерфейс и значение C-VID пользователя (или же S-VID, если трафик поступает через промежуточную сеть PB). В нашем примере таким способом задано отображение для сервиса с I-SID 56, который монопольно использует интерфейсы коммутаторов BEB1 и BEB2, не разделяя их с другими сервисами. В терминологии MEF это сервис EPL (а тип сервиса — E-LINE).

В том случае, когда на один и тот же интерфейс поступает трафик более чем одного сервиса, при отображении нужно также учитывать значение C-VID (или S-VID, если трафик принимается от сети PB). Этот случай имеет место для сервисов с I-SID 144 и 108, так как они разделяют один и тот же интерфейс коммутаторов BEB1 и BEB2. Поэтому такие отображения нужно конфигурировать с учетом значений C-VID; например, если клиент использует для значения C-VID 305 и 500 для маркировки трафика двух различных услуг, то C-VID 305 отображается на I-SID 144, а C-VID 500 — на I-SID 108.

В терминологии MEF сервис с I-SID 144 является сервисом EVPL (тип E-LINE), а сервис с I-SID 108 — сервисом EVP-LAN (тип E-LAN).

Пользовательские MAC-адреса

Теперь нам нужно рассмотреть важный вопрос применения пользовательских MAC-адресов. Магистральным коммутаторам сети PBB знание пользовательских адресов не требуется, так как они передают кадры только на основании комбинации B-MAC/B-VID.

А вот поведение пограничных коммутаторов в отношении пользовательских MAC-адресов зависит от типа сервиса.

При отображении кадров сервиса типа E-LINE (то есть «точка-точка») на определенное соединение I-SID пограничные коммутаторы не применяют пользовательские MAC-адреса, так как все кадры, независимо от их адресов назначения, передаются одному и тому же выходному пограничному коммутатору. Например, для сервисов с I-SID 56 и 144 коммутатор BEB1 всегда задействует MAC-адрес коммутатора BEB2 в качестве B-MAC DA при формировании несущего (нового) кадра, который переносит инкапсулированный пользовательский кадр через сеть PBB.

Однако при отображении кадров сервисов типа E-LAN и E-TREE (то есть «многоточка-многоточка») у входного коммутатора всегда существует несколько выходных пограничных коммутаторов, поддерживающих этот сервис. Например, у входного коммутатора BEB1 при обслуживании кадров сервиса с I-SID 108 есть альтернатива — отправить пришедший кадр коммутатору BEB2 или BEB3.

Для принятия решения в таких случаях применяются пользовательские MAC-адреса. Пограничные коммутаторы, поддерживающие сервисы типа E-LAN и E-TREE, изучают пользовательские MAC-адреса и посылают кадр выходному коммутатору, связанному с той сетью пользователя, в которой находится MAC-адрес назначения C-MAC DA.

Так, в нашем примере коммутатор BEB1 изучает адреса C-MAC SA кадров, поступающих по I-SID 108, чтобы знать, подключены ли узлы с этими адресами к BEB2 или BEB3. В результате BEB1 создает таблицу продвижения (табл. 21.1).

Таблица 21.1. Таблица продвижения для сервиса I-SID 108

C-MAC	I-SID	B-MAC	B-VID
C-MAC-1	108	B-MAC-2	1033
C-MAC-2	108	B-MAC-2	1033
C-MAC-3	108	B-MAC-3	1033
C-MAC-4	108	B-MAC-3	1033
...	108	...	1033

На основании этой таблицы коммутатор BEB1 по адресу назначения C-MAC выбирает соответствующий адрес выходного пограничного коммутатора и помещает его в формируемый кадр, например, для кадра с адресом назначения C-MAC-2 это будет B-MAC-2. В том же случае, когда пользовательский адрес назначения еще не изучен, коммутатор BEB1 помещает в поле B-MAC широковещательный адрес. Таким же образом обрабатываются кадры с широковещательным пользовательским адресом.

Инжиниринг трафика и отказоустойчивость

Возможности инжиниринга трафика в сетях PBB ограничены функциональностью протокола STP, который остается и в этом типе сетей основным протоколом, обеспечивающим отказоустойчивость сети при наличии избыточных связей. Этот протокол не дает администратору полного контроля над путями передачи трафика, хотя, как вы знаете из главы 14, некоторые возможности подобного рода у него имеются, так как администратор может влиять на выбор покрывающего дерева за счет назначения приоритетов коммутаторам и их портам. Применение протокола MSTP дает дополнительные возможности устанавливать в сети различные покрывающие деревья для различных виртуальных локальных сетей — это свойство использовано в сети, показанной на рис. 21.13.

Так как кадры протокола STP сети провайдера и сетей клиентов в технологии PBB изолированы друг от друга, то здесь нет необходимости применять различные групповые адреса для коммутаторов провайдера и клиентов, как это сделано в стандарте PB.

Ограниченные возможности стандарта PBB в отношении инжиниринга трафика преодолены в стандарте PBB TE, но только для случая двухточечных соединений, то есть для услуг типа E-LINE.

Магистральные мосты провайдера с поддержкой инжиниринга трафика

Технология PBB TE (Provider Backbone Bridge Traffic Engineering — магистральные мосты провайдера с поддержкой инжиниринга трафика) ведет свое начало от фирменной технологии PBT (Provider Backbone Transport — магистральный транспорт провайдера) компании Nortel. В начале 2007 года для стандартизации этой технологии была образована рабочая группа IEEE 802.1Qay, работа которой на момент написания данной книги еще не была завершена (ее окончание планировалось на конец 2009 года).

Технология PBB TE базируется на технологии PBB, в ней используется та же самая схема инкапсуляции кадров и отображения пользовательских соединений на провайдерские туннели.

Главными целями разработчиков технологии PBB TE были:

- поддержка функций инжиниринга трафика для магистральных виртуальных локальных сетей (B-VLAN) с топологией «точка-точка». (эти сети часто называют транками, или туннелями);
- обеспечение «быстрой» отказоустойчивости со скоростью, сравнимой со скоростью работы защиты соединений в технологии SDH.

Поставленные цели достигаются в технологии PBB TE за счет следующих изменений технологии PBB и классической технологии локального моста:

- Отключение протокола STP.
- Отключение механизма автоматического изучения магистральных MAC-адресов.
- Использование пары «B-VID/B-MAC-DA» в качестве метки туннеля. В принципе любой коммутатор, который поддерживает технику VLAN (стандарт IEEE 802.1Q), продвигает кадры на выходной порт, анализируя два указанных в кадре значения: MAC-адрес назначения и номер VLAN. Поэтому данное свойство просто предполагает, что коммутатор ведет себя в соответствии с алгоритмом продвижения, описанным в стандарте 802.1Q, но только для магистральных адресов и магистральных виртуальных локальных сетей.
- Предварительная прокладка первичного (основного) и резервного туннеля для тех случаев, когда нужно обеспечить отказоустойчивость туннеля.
- Описанные первые три свойства технологии PBB TE позволяют администратору или системе управления сетью формировать пути прохождения через сеть произвольным образом, независимо от того, имеют ли они минимальную метрику до некоторого коммутатора, названного корневым, или нет — то есть обеспечивают поддержку функций инжиниринга трафика. Пара «B-VID/B-MAC-DA» является аналогом метки пути LSP технологии MPLS, однако в отличие от метки MPLS значение этой пары остается неизменным в процессе перемещения кадра по сети провайдера.

Посмотрим, как работает технология PBB TE, на примере сети, изображенной на рис. 21.14.

В этой сети сконфигурировано два туннеля:

- Основной туннель с B-VID 1007 между BEB1 и BEB2, проходящий через BCB2 и BCB5. Нужно отметить, что в отличие от туннелей MPLS туннели PBB TE являются двунаправленными.
- Резервный туннель с B-VID 1033, соединяющий те же конечные точки BEB1 и BEB2, но проходящий через другие промежуточные коммутаторы BCB1 и BCB4, что позволяет обеспечить работоспособность резервного туннеля при отказе какого-либо элемента (коммутатора или линии связи) основного туннеля.

Организация обоих туннелей достигается путем ручного конфигурирования таблиц продвижения во всех коммутаторах сети, через которые проходят туннели. Например, таблица продвижения коммутатора BEB1 после такого конфигурирования выглядит так, как показано в табл. 21.2.

Для устойчивой работы сети PBB TE необходимо, чтобы комбинация B-VID/B-MAC-DA была уникальной в пределах этой сети. Уникальность может обеспечиваться разными способами. Если в качестве адресов B-MAC-DA в таблицах продвижения указываются адреса физических интерфейсов коммутаторов, то уникальность обеспечивается традиционным способом — за счет централизованной схемы назначения значения старших трех байтов этих адресов, представляющих собой уникальный идентификатор производителя оборудования OUI (как вы знаете из главы 12, эту схему контролирует комитет IEEE 802).

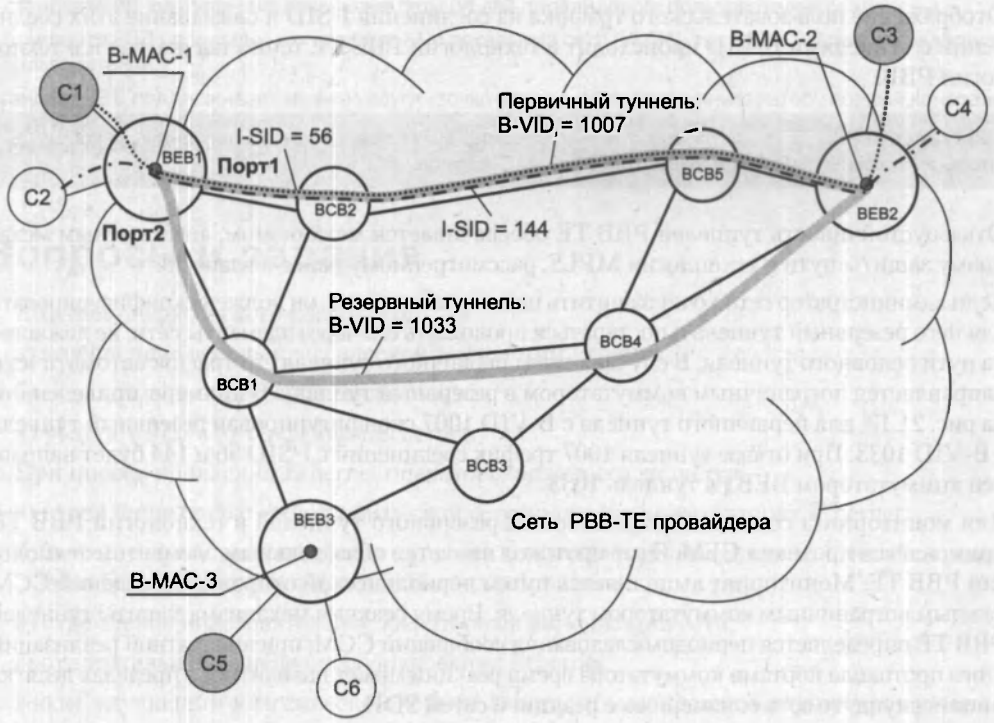


Рис. 21.14. Организация услуг в сети PBB TE

Таблица 21.2. Таблица продвижения коммутатора BEB1

MAC-адрес назначения (B-MAC-DA)	VLAN ID (B-VID)	Выходной порт
B-MAC-2	1007	Port1
B-MAC-2	1033	Port2
...		

Существует также практика ручного назначения коммутаторам так называемых MAC-адресов обратной связи, которые относятся не к отдельному физическому интерфейсу, а к коммутатору в целом. Такие адреса удобно использовать для организации туннелей между устройствами, так как конфигурация туннеля не связана непосредственно с данным коммутатором и остается неизменной при его замене. При ручном задании MAC-адресов ответственность за их уникальность лежит на администраторе; понятно, что такое решение может работать только в пределах одного административного домена.

Добавление значения B-VID к адресу B-MAC-DA позволяет организовать к одному и тому же пограничному коммутатору до 1024 туннелей с различными в общем случае путями прохождения через сеть. Это дает администратору или системе управления широкие возможности в отношении инжиниринга трафика в сетях PBB TE.

Нужно подчеркнуть, что таблицы продвижения в сети PBB TE имеют стандартный вид (для коммутаторов, поддерживающих технику VLAN). Изменяется только способ построения этих таблиц — вместо автоматического построения на основе изучения адресов передаваемых кадров имеет место их внешнее формирование.

Отображение пользовательского трафика на соединения I-SID и связывание этих соединений с туннелями B-VID происходит в технологии PBB TE точно так же, как и в технологии PBB.

Так как сети PBB TE поддерживают только соединения «точка-точка», то пограничные коммутаторы не должны изучать пользовательские MAC-адреса.

Отказоустойчивость туннелей PBB TE обеспечивается механизмом, аналогичным механизму защиты пути в технологии MPLS, рассмотренному ранее в главе 20.

Если администратор сети хочет защитить некоторый туннель, он должен сконфигурировать для него резервный туннель и постараться проложить его через элементы сети, не лежащие на пути основного туннеля. В случае отказа первичного туннеля его трафик автоматичски направляется пограничным коммутатором в резервный туннель. В примере, приведенном на рис. 21.13, для первичного туннеля с B-VID 1007 сконфигурирован резервный туннель с B-VID 1033. При отказе туннеля 1007 трафик соединений с I-SID 56 и 144 будет направлен коммутатором BEB1 в туннель 1033.

Для мониторинга состояний первичного и резервного туннелей в технологии PBB TE применяется протокол CFM. Этот протокол является обязательным элементом технологии PBB TE. Мониторинг выполняется путем периодической отправки сообщений CCM каждым пограничным коммутатором туннеля. Время реакции механизма защиты туннелей PBB TE определяется периодом следования сообщений CCM; при аппаратной реализации этого протокола портами коммутатора время реакции может находиться в пределах десятка миллисекунд, то есть соизмеримо с реакцией сетей SDH.

Выводы

В наиболее широком смысле под Ethernet операторского класса понимают как услуги Ethernet, которые операторы связи предоставляют в глобальном масштабе, так и технологии, на основе которых эти услуги организуются.

Движущими силами превращения Ethernet в технологию операторского класса являются:

- привлекательность для пользователей услуг Ethernet в глобальном масштабе;
- низкая стоимость оборудования Ethernet;
- унификация технологий канального уровня.

Существует несколько вариантов организации глобальной услуги Ethernet:

- Ethernet поверх MPLS (EoMPLS);
- Ethernet поверх Ethernet;
- Ethernet поверх транспорта первичных сетей.

Основные потребительские свойства глобальной услуги Ethernet стандартизованы форумом MEF.

В технологии EoMPLS применяется двухуровневая иерархия соединений: на нижнем уровне работают туннели MPLS, а на верхнем — псевдоканалы, переносящие пользовательский трафик.

С помощью технологии EoMPLS провайдер может оказывать услуги двух типов: VPWS (соединения «точка-точка») и VPLS (соединения «каждый с каждым»).

Для реализации варианта Ethernet поверх Ethernet комитет IEE802.1 разработал три стандарта:

- мосты провайдера (PB);
- магистральные мосты провайдера (PBB);
- магистральные мосты провайдера с поддержкой инжиниринга трафика (PBB).

В стандарте PB виртуальные локальные сети (VLAN) провайдера и пользователей разделены.

В стандарте PBB разделены как виртуальные локальные сети (VLAN), так и MAC-адреса провайдера и пользователей.

Стандарт PBB поддерживает только услуги «точка-точка», но дает администратору полный контроль над путями следования трафика через сеть. Еще одним важным новым свойством этого стандарта является механизм быстрой защиты пользовательских соединений.

Вопросы и задания

1. Ethernet операторского класса это:
   а) улучшенная версия классической технологии Ethernet;
   б) новая услуга операторов связи;
   в) услуга VPLS с интерфейсом Ethernet.
2. Причинами появления Ethernet операторского класса является:
   а) стремление операторов строить свои сети только на коммутаторах Ethernet;
   б) желание пользователей объединять свои территориально распределенные сайты, «как если бы они принадлежали одной локальной сети»;
   в) стремление пользователей и операторов к унификации сети;
   г) относительная дешевизна оборудования Ethernet.
3. Какие улучшения классической версии Ethernet были сделаны для превращения ее в технологию операторского класса? Варианты ответов:
   а) повышена надежность оборудования Ethernet;
   б) улучшены эксплуатационные свойства оборудования Ethernet;
   в) добавлена возможность изоляции адресных пространств клиентов и оператора.
4. Чем вариант «Ethernet поверх MPLS» отличается от варианта «Ethernet поверх транспорта»? Варианты ответов:
   а) характеристиками предоставляемой услуги;
   б) используемой внутренней транспортной технологией для предоставления одной и той же услуги;
   в) в первом случае в сети оператора используется техника коммутации пакетов, во втором — коммутации каналов.
5. Что стандартизуют спецификации форума MEF? Варианты ответов:
   а) топологию связей услуги Ethernet;
   б) возможность использования идентификаторов VLAN для определения топологии связей услуги;
   в) параметры пропускной способности соединений.
6. Псевдоканал MPLS это:
   а) путь LSP второго уровня иерархии;
   б) эмулятор некоторого телекоммуникационного сервиса;
   в) путь LSP первого уровня иерархии.

7. Какое максимальное количество псевдоканалов можно проложить в одном туннеле MPLS?
8. Должно ли устройство PE изучать MAC-адреса клиентов при оказании услуги VPWS?
9. Виртуальный коммутатор услуги VPLS изучает MAC-адреса, приходящие:

   а) по логическому интерфейсу; б) по псевдоканалам.
10. С какой целью для сообщений CCM введено понятие уровня? Варианты ответов:

    а) для мониторинга иерархических многоуровневых соединений MPLS;

    б) для мониторинга многодоменных сетей Ethernet;

    в) для обеспечения приоритетности тестирования сети оператора связи.
11. Верно ли утверждение «Стандарт Y.1731 дополняет функции стандарта CFM набором функции мониторинга производительности сети»?
12. Стандарт «Мосты провайдера» обеспечивает изоляцию:

    а) виртуальных локальных сетей клиентов и провайдера;

    б) MAC-адресов клиентов и провайдера;

    в) MAC-адресов пограничных и магистральных коммутаторов провайдера.
13. Пограничные коммутаторы провайдера, работающие в соответствии со стандартом «Магистральные мосты провайдера», должны изучать MAC-адреса клиентов:

    а) всегда; б) никогда; в) при оказании услуги E-LAN.

ГЛАВА 22 Удаленный доступ

Термин «удаленный доступ» (remote access) часто употребляют, когда речь идет о доступе пользователя домашнего компьютера к Интернету или сети предприятия, которая находится от него на значительном расстоянии, означающем необходимость применения глобальных связей. В последнее время под удаленным доступом стали понимать не только доступ изолированных компьютеров, но и домашних сетей, объединяющих несколько компьютеров членов семьи. Такими же небольшими сетями располагают малые офисы предприятий, насчитывающие 2–3 сотрудника.

Организация удаленного доступа является одной из наиболее острых проблем компьютерных сетей в настоящее время. Она получила название «проблемы последней мили», где под последней милей подразумевается расстояние от точки присутствия (POP) оператора связи до помещений клиентов. Сложность этой проблемы определяется несколькими факторами. С одной стороны, современным пользователям необходим высокоскоростной доступ, обеспечивающий качественную передачу трафика любого типа, в том числе данных, голоса, видео. Для этого нужны скорости в несколько мегабит, а для качественного приема телевизионных программ — в несколько десятков мегабит в секунду. С другой стороны, подавляющее большинство домов в больших и малых городах и особенно в сельской местности по-прежнему соединены с точками присутствия операторов связи абонентскими окончаниями телефонной сети, которые не были рассчитаны на передачу компьютерного трафика.

Кардинальная перестройка кабельной инфраструктуры доступа требует времени — слишком масштабна эта задача из-за огромного количества зданий и домов, географически рассеянных по огромной территории. И хотя в некоторых странах в последнее время стали прокладывать к домам высокоскоростные оптические линии, таких стран не так уж много, да и этот процесс затронул пока только большие города и крупные здания с множеством потенциальных пользователей.

Долгое время наиболее распространенной технологией доступа был коммутируемый доступ, когда пользователь устанавливал коммутируемое соединение с корпоративной сетью или Интернетом через телефонную сеть с помощью модема, работающего в голосовой полосе частот. Такой способ обладает существенным недостатком — скорость доступа ограничена несколькими десятками килобит в секунду из-за фиксированной узкой полосы пропускания примерно в 3,4 кГц, выделяемой каждому абоненту телефонной сети (вспомните технику мультиплексирования, применяемую в телефонных сетях и описанную в главе 9). Такие скорости сегодня устраивают все меньше и меньше пользователей.

Для организации скоростного удаленного доступа сегодня привлекаются различные технологии, в которых используется только существующая инфраструктура абонентских окончаний — телефонные сети или сети кабельного телевидения. После достижения POP поставщика услуг по такому окончанию компьютерные данные уже не следуют по телефонной сети или сети кабельного телевидения, а ответвляются с помощью специального оборудования в сеть передачи данных. Это позволяет преодолеть ограничения на полосу пропускания, отводимую абоненту в телефонной сети или сети кабельного телевидения, и повысить скорость доступа. Наиболее популярными технологиями такого типа являются технология ADSL, использующая телефонные абонентские окончания, и кабельные модемы, работающие поверх сети кабельного телевидения. Эти технологии обеспечивают скорость от нескольких сотен килобит до нескольких десятков мегабит в секунду.

Применяются также различные беспроводные технологии доступа, обеспечивающие как фиксированный, так и мобильный доступ. Набор таких беспроводных технологий очень широк, в него входят и беспроводные сети Ethernet (802.11), различные фирменные технологии, передача данных по сети мобильной телефонии, а также технологии фиксированного доступа, например, стандарта 802.16.

В этой главе мы рассмотрим основные схемы и наиболее популярные технологии удаленного доступа.

Схемы удаленного доступа

Рисунок 22.1 иллюстрирует разнообразный и пестрый мир удаленного доступа. Мы видим здесь клиентов различных типов, отличающихся используемым оборудованием и требованиями к параметрам доступа. Кроме того, помещения клиентов могут быть соединены с ближайшей точкой доступа оператора связи (то есть с ближайшим центральным офисом, если пользоваться терминологией операторов телефонной сети) различными способами: с помощью аналогового или цифрового окончания телефонной сети, телевизионного кабеля, беспроводной связи. Наконец, сам оператор связи может иметь различную специализацию, то есть быть либо поставщиком телефонных услуг, либо поставщиком услуг Интернета, либо оператором кабельного телевидения или же быть универсальным оператором, предоставляющим весь спектр услуг и обладающим собственными сетями всех типов.

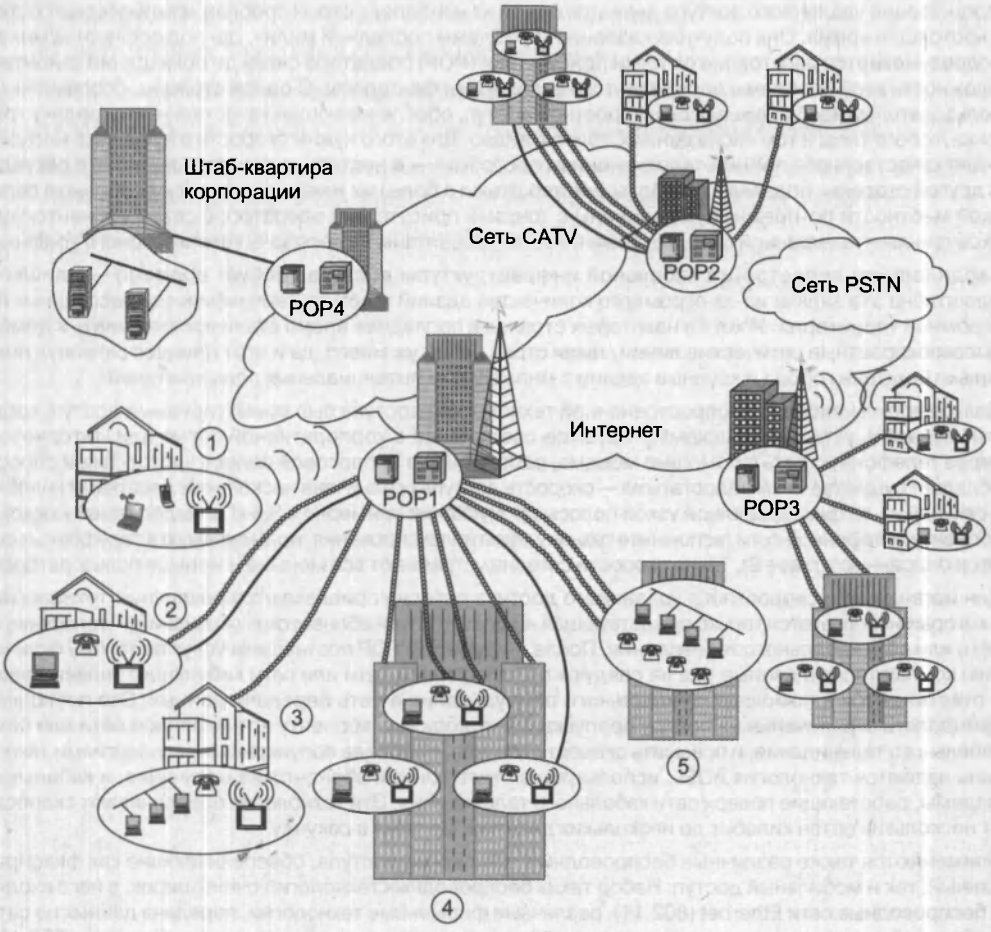


Рис. 22.1. Клиенты удаленного доступа

Типы клиентов и абонентских окончаний

Рассмотрим каждый элемент схемы доступа, показанный на рис. 22.1, более подробно.

Клиенты 1 и 2 являются наиболее типичными пользователями, так как каждый из них имеет только один компьютер, которому необходимо обеспечить доступ к удаленной компьютерной сети. Помимо компьютера эти клиенты пользуются телефоном и телевизором, поэтому абонентские окончания этих устройств можно использовать для организации доступа компьютера к сети передачи данных.

Клиент 2 пользуется двумя кабельными абонентскими окончаниями, традиционным аналоговым телефонным на основе витой пары и коаксиальным телевизионным кабелем кабельного телевидения. Эти абонентские окончания обладают существенно разными характеристиками. Так, витая пара при расстоянии 1–2 км между помещением клиента и POP поставщика услуг обычно имеет полосу пропускания порядка нескольких мегагерц, в то время как коаксиальный кабель обеспечивает полосу пропускания в несколько десятков мегагерц.

У клиента 1 отсутствуют проводные абонентские окончания, так как он пользуется мобильным телефоном, кроме того, он не является клиентом кабельного телевидения, принимая телевизионный сигнал только по воздуху.

Таким образом, для организации удаленного доступа для клиента 2 поставщик услуг может использовать либо существующее телефонное абонентское окончание, либо телевизионный кабель. Для клиента 1 такой возможности нет, поэтому поставщик услуг должен предоставить ему беспроводную связь или же проложить новый кабель между его домом и ближайшей точкой присутствия.

Отличительной особенностью клиентов 1 и 2 является несимметричный характер трафика, так как домашние пользователи в основном загружают информацию на свой компьютер в процессе путешествий по Интернету. Ответом на такие потребности являются асимметричные технологии, такие как ADSL.

Клиент 3 отличается от двух предыдущих тем, что имеет несколько компьютеров, объединенных в локальную сеть. Таким клиентом может быть как частное лицо, так и небольшой офис. Удаленный доступ для локальной сети отличается повышенными требованиями к пропускной способности. Кроме того, трафик может иметь симметричный характер, если домашняя сеть включает сервер, поставляющий информацию пользователям Интернета или сотрудникам других офисов предприятия. Так как клиент 3 не имеет кабельного окончания сети CATV (cable TV), то ему можно обеспечить доступ только по телефонному окончанию. Клиент 3 может организовать свою IP-сеть различными способами. Он может попросить у поставщика услуг пул IP-адресов, так чтобы каждый его компьютер имел отдельный публичный постоянный IP-адрес. Это наиболее гибкий вариант для клиента, так как в этом случае каждый его компьютер может быть полноправным узлом Интернета и исполнять роль не только клиентской машины, но и сервера с зарегистрированным доменным именем. Очевидно, что в этом случае локальная сеть клиента должна иметь пограничный маршрутизатор, через который осуществлять связь с сетью поставщика услуг. Другой вариант организации IP-сети может быть основан на использовании техники NAT, описанной в главе 18.

Клиенты 4 являются жителями многоквартирного дома, который соединен с POP многочисленными витыми парами телефонных абонентских окончаний (по одной для каждой квартиры), а также кабелем CATV. Использование одного кабеля CATV для большого количества клиентов порождает дополнительные проблемы при организации доступа, так как

кабель в этом случае является разделяемой средой. Применение телефонных абонентских окончаний для удаленного доступа жителей многоквартирного дома ничем не отличается от подключения отдельного абонента (клиента 2). И хотя большая часть жильцов дома использует обычные аналоговые телефонные окончания, жильцы нескольких квартир — абоненты сети ISDN, окончания которой являются цифровыми (при том, что они, так же как и аналоговые телефонные окончания, работают на витой паре). Хотя сеть ISDN была разработана как универсальная, то есть предоставляющая наряду с сервисами телефонии и сервисы передачи данных, на практике она используется как обычная телефонная сеть.

Клиенты 5 также являются жильцами многоквартирного дома, но в этом доме поставщик услуг развернул локальную сеть. К этой локальной сети подключаются компьютеры тех жильцов дома, которые решили стать абонентами данного поставщика услуг. Такой вариант эффективен для поставщика услуг при достаточно большом количестве абонентов в доме. Локальная сеть многоквартирного дома требует более высоких скоростей доступа, чем отдельные компьютеры или домашние сети индивидуальных клиентов, поэтому поставщик услуг должен использовать абонентское окончание с широкой полосой пропускания — для этой цели может быть применен существующий кабель CATV, специально проложенный коаксиальный кабель Ethernet или также заново проложенный оптический кабель.

Поставщик услуг удаленного доступа может обслуживать клиентов всех типов или же специализироваться на каком-то определенном типе клиентов, например жителях частных или многоквартирных домов, работниках небольших офисов. Универсальный поставщик услуг доступа должен поддерживать любые варианты организации «последней мили», что усложняет его оборудование и применяемые технологии доступа.

В любом случае, для передачи данных по какому-либо абонентскому окончанию поставщик услуг должен обеспечить передачу через это окончание компьютерных данных и совместить эту передачу с передачей информации, для которой это окончание было спроектировано, например с аналоговой телефонной информацией или с сигналом кабельного телевидения. Затем на основе этих средств физического уровня поставщик услуг должен предоставить клиенту тот или иной вариант сервиса доступа.

Еще одной проблемой, которую должен решить оператор, является организация доступа клиентов, которые физически подключены к абонентским окончаниям *других* поставщиков услуг связи. Так, пусть на рисунке POP1 и POP2 принадлежат поставщику *A*, а POP3 — поставщику *B*. Для того чтобы поставщик *A* мог предоставлять услуги доступа к сети передачи данных клиентам, подключенным к POP3, у него должно быть заключено соответствующее соглашение с поставщиком *B*. Это соглашение может регламентировать различные способы взаимодействия поставщиков услуг, которые мы уже обсуждали в главе 5. Например, поставщик услуг *A* может арендовать у поставщика услуг *B* те абонентские окончания, которыми пользуются его клиенты, с тем чтобы затем передавать получаемые по ним данные в свою сеть и направлять их далее в соответствии с потребностями клиентов. В другом случае абонентские окончания могут оставаться в распоряжении поставщика услуг *B*, который должен отделять поступающие компьютерные данные от телефонной или телевизионной информации и направлять в сеть поставщика услуг *A*. Очевидно, что между сетями передачи данных поставщиков услуг *A* и *B* должно поддерживаться взаимодействие.

Наиболее простой вариант доступа в Интернет предоставляет клиенту **незащищенное соединение** с серверами корпоративной сети, однако такое соединение грозит плохими последствиями. Во-первых, конфиденциальные данные, передаваемые по Интернету, могут

быть перехвачены или искажены. Во-вторых, при таком способе администратору корпоративной сети трудно ограничить доступ к своей сети несанкционированных пользователей, так как IP-адреса легальных пользователей (сотрудников предприятия) заранее неизвестны. Поэтому предприятия предпочитают безопасный доступ, основанный на технологии защищенного канала. Эта технология рассматривается в главе 24.

Мультиплексирование информации на абонентском окончании

Как мы видим на рис. 22.1, большинство домов и многоквартирных зданий связаны с POP либо телефонными абонентскими окончаниями, либо абонентскими окончаниями кабельного телевидения.

Поэтому для обеспечения клиентов *трем* основными на сегодня видами доступа (к телефонной сети, телевизионной сети и сети передачи данных) необходимо реализовать одновременную передачу информации разного типа по одной линии связи. Например, совместить передачу данных с передачей голоса и по телефонному окончанию или же совместить передачу данных с передачей телевизионного сигнала по коаксиальному кабелю.

В идеале желательно использовать единственное абонентское окончание, способное передавать информацию всех трех типов. К сожалению, витая пара на эту роль не подходит, так как ее полоса пропускания на расстояниях в несколько километров не превышает 1 МГц. Этого явно недостаточно для одновременной передачи голоса, компьютерных данных со скоростями в несколько мегабит в секунду и цветного телевизионного изображения.

Поэтому на роль консолидирующего абонентского окончания могут претендовать только коаксиальный кабель сети CATV и широкополосные беспроводные линии связи. Естественно, мы имеем в виду уже существующие и широко распространенные типы абонентских окончаний. Если же говорить о прокладке нового кабеля, что актуально в основном для новых крупных зданий, то к этому списку нужно добавить оптический кабель.

Почти во всех технологиях доступа, которые мы будем рассматривать в следующих разделах, требуется мультиплексирование каких-либо двух или всех трех упомянутых типов информации на абонентском окончании. Так, в линии ADSL аналоговые телефонные окончания служат для мультиплексирования голоса и компьютерных данных, кабельные модемы совмещают передачу телевизионного изображения и компьютерных данных по коаксиальному кабелю. Существуют также различные технологии беспроводного доступа, которые обеспечивают передачу телевизионного сигнала и компьютерных данных, а иногда и телефонии в одном абонентском окончании. Исключением является только наиболее старая технология доступа, а именно коммутируемый доступ, при котором аналоговое абонентское окончание может использоваться телефоном или модемом компьютера только попеременно.

Схема организации доступа с помощью универсального абонентского окончания показана на рис. 22.2.

Наиболее часто для мультиплексирования информации в абонентском окончании применяется техника FDM. Каждому из трех типов информации выделяется определенная полоса частот, ширина которой соответствует потребностям абонента. Для телефонного

соединения выделяется полоса 4 КГц, соответствующая стандартной полосе абонента аналоговых телефонных сетей. Компьютерным данным нужна более широкая полоса, при асимметричном доступе для преобладающего нисходящего (входящего) трафика нужно выделить полосу, как минимум, в несколько сотен килогерц, а лучше — в несколько мегагерц. Менее интенсивный восходящий (выходящий) трафик требует полосы в несколько десятков килогерц. В кабельном телевидение традиционно используются полосы по 6 МГц для каждого абонента, но при этом передается только нисходящий трафик.

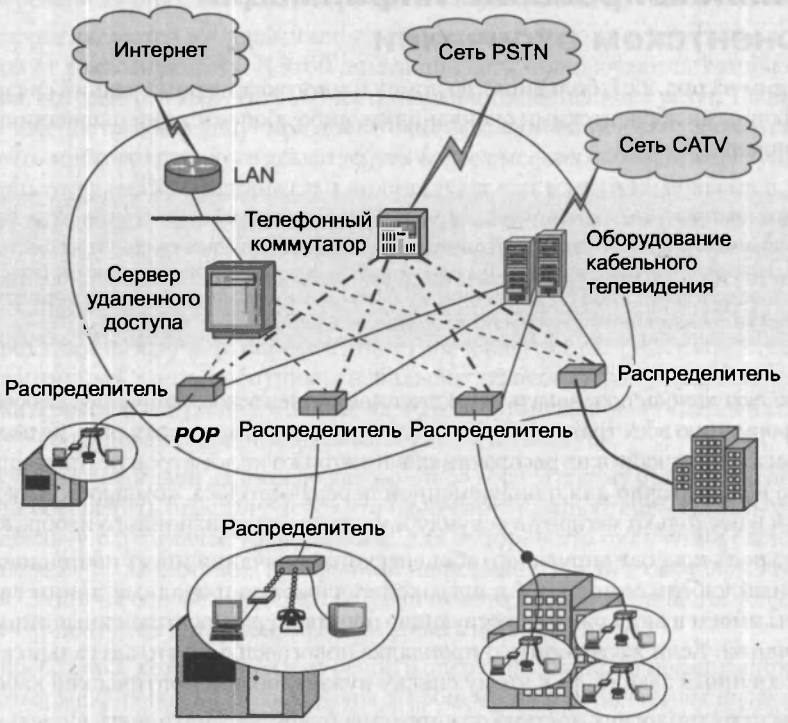


Рис. 22.2. Мультиплексирование трех типов информации в абонентских окончаниях

Для того чтобы реализовать выбранную схему FDM, в помещении клиента и точках присутствия устанавливаются **распределители**, которые выполняют операции мультиплексирования и демультиплексирования сигналов. Распределитель чаще всего представляет собой пассивный фильтр, который выделяет нужные диапазоны частот и передает каждый диапазон на отдельный выход. К выходу распределителя подключаются терминальные устройства абонента — телефон, телевизор и компьютер. Так как компьютер использует дискретные сигналы для обмена данными, то для него требуется дополнительное устройство, которое будет преобразовывать дискретные сигналы в аналоговые сигналы необходимого диапазона частот.

Большинство пользователей привыкли иметь дело с **коммутируемыми (телефонными) модемами**, которые работают со стандартной полосой 4 кГц аналоговых телефонных сетей. Телефонные модемы не разделяют эту полосу с другими устройствами, целиком занимая ее для передачи компьютерных данных. Очевидно, что распределитель в этом случае не нужен.

Существуют также **устройства ADSL** и **кабельные модемы**; первые работают на абонентских окончаниях телефонных сетей, а вторые — на кабелях CATV. Для этих окончаний распределитель необходим, так как по ним вместе с компьютерными данными передается и основная для них информация, телефонная или телевизионная.

В POP поставщика услуг каждое абонентское окончание также подключено к распределителю, который выполняет аналогичные операции мультиплексирования и демультиплексирования на другом конце кабеля. В результате телефонная информация поступает с телефонных выходов распределителя на телефонный коммутатор поставщика услуг, который передает ее в телефонную сеть. Телевизионные сигналы от соответствующих выходов распределителя собираются на оборудовании CATV, которое может быть связано с сетью CATV этого поставщика услуг.

И, наконец, компьютерные данные поступают на устройство, концентрирующее компьютерный трафик и передающее его в локальную сеть поставщика услуг. Это устройство называют по-разному, на рисунке можно видеть одно из популярных названий — **сервер удаленного доступа** (Remote Access Server, RAS). Можно встретить и другие названия, например **концентратор удаленного доступа** (Remote Access Concentrator, RAC), **мультиплексор доступа** или **терминальная система**. Будем для определенности называть здесь такое устройство сервером удаленного доступа. Обычно оно содержит большое количество модемов, которые выполняют обратные операции по отношению к модемам пользователей, то есть модулируют нисходящий трафик и демодулируют восходящий. Помимо модемов, RAS включает маршрутизатор, который собирает трафик от модемов и передает его в локальную сеть POP. Из этой локальной сети трафик передается обычным способом в Интернет или в определенную корпоративную сеть.

Мы рассмотрели обобщенную схему доступа, которая в зависимости от выбранного типа абонентского окончания и типа модема требует различных технологий доступа. Нужно подчеркнуть, что в терминах модели OSI все они являются технологиями физического уровня, так как создают поток битов между компьютером клиента и локальной сетью поставщика услуг. Для работы протокола IP поверх этого физического уровня должен использоваться один из протоколов канального уровня. Сегодня наиболее часто при удаленном доступе применяется протокол PPP, который поддерживает такие важные функции, как назначение IP-адреса клиентскому компьютеру, а также аутентификацию пользователя.

Режим удаленного узла

Наиболее распространенной услугой сегодня является предоставление **доступа к общедоступному домену Интернета**. При этом подразумевается, что поставщик услуг обеспечивает маршрутизацию IP-трафика между компьютером и любым сайтом Интернета, имеющим публичный адрес (или же имеющим частный адрес и обеспечивающим публичный доступ посредством техники NAT). Когда клиент располагает одним компьютером, для предоставления такой услуги поставщик услуг обычно использует режим удаленного узла.

Режим удаленного узла позволяет компьютеру клиента стать узлом удаленной локальной сети, что означает для его пользователя возможность получения всего спектра услуг обычного пользователя узла, физически расположенного в локальной сети.

Для этого поставщик услуг резервирует для своих клиентов удаленного доступа пул IP-адресов из диапазона адресов одной из своих подсетей. Для тех клиентов, которые не

нуждаются в постоянном доступе к Интернету, услуга предоставляется как коммутируемая, и IP-адрес им назначается динамически только на время подключения клиента. Режим удаленного узла позволяет экономить адреса подсетей, так как в стандартном режиме IP-маршрутизатор должен назначить каждому своему порту адрес отдельной подсети, что для одного узла, из которого состоят сети многих клиентов, явно избыточно. Для тех же клиентов, которым требуется постоянное соединение, адрес может назначаться как на постоянной основе, так и динамически на время активности клиента.

Для обеспечения режима удаленного узла RAS поставщика услуг поддерживает *протокол Proxy-ARP*, рассмотренный в главе 15. Эта особенность отличает сервер удаленного доступа от обычного IP-маршрутизатора (рис. 22.3).

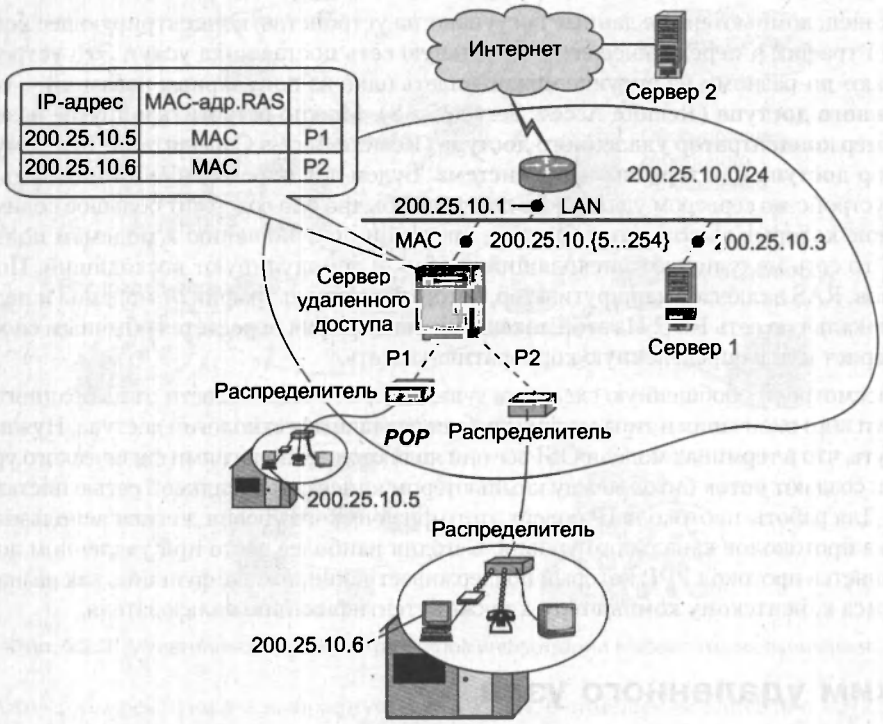


Рис. 22.3. Использование протокола Proxy-ARP при организации удаленного доступа

Для удаленных узлов в локальной сети поставщика услуг, имеющей адрес 200.25.10.0/24, выделен пул адресов от 200.25.10.5 до 200.25.10.254. Если клиент пользуется коммутируемым сервисом, то при его соединении с сетью поставщика услуг (например, по протоколу PPP), ему *временно* назначается адрес из этого пула. Так, компьютеру первого клиента был назначен адрес 200.25.10.5, а компьютеру второго клиента — адрес 200.25.10.6. При подключении к сети этих удаленных узлов сервер удаленного доступа заносит в специальную таблицу, являющуюся аналогом ARP-таблицы, следующие записи:

200.25.10.5 — MAC — P1

200.25.10.6 — MAC — P2

Здесь MAC обозначает адрес внутреннего интерфейса сервера удаленного доступа, а P1 и P2 — номера портов, к которым подключены клиенты удаленного доступа.

Если, например, сервер 2 (см. рис. 22.3), подключенный к сети одного из поставщиков услуг, посылает пакет компьютеру первого клиента, то маршрутизатор поставщика услуг считает, что пакет направлен к одному из узлов, принадлежащих непосредственно присоединенной подсети 200.25.10.0/24. Поэтому маршрутизатор посылает ARP-запрос, содержащий адрес 200.25.10.5. На этот запрос отвечает не компьютер первого клиента, а RAS, сообщая в ARP-ответе маршрутизатору *собственный* MAC-адрес. После этого маршрутизатор направляет IP-пакет, упакованный в кадр Ethernet с MAC-адресом RAS. RAS извлекает IP-пакет из пришедшего кадра Ethernet и по IP-адресу определяет в таблице номер порта, на который ему нужно направить пакет. В данном случае это порт P1. RAS инкапсулирует пакет в кадр PPP, используемый для работы на абонентском окончании, соединяющем RAS с компьютером первого клиента.

В том случае, когда у клиента имеется своя локальная сеть, узлы которой имеют зарегистрированные публичные IP-адреса, RAS работает как обычный маршрутизатор, и такой режим уже не называют режимом удаленного узла.

Режим удаленного управления и протокол telnet

Режим удаленного управления, называемый также режимом **терминального доступа**, предполагает, что пользователь превращает свой компьютер в виртуальный терминал другого компьютера, к которому он получает удаленный доступ.

В период становления компьютерных сетей, то есть в 70-е годы, поддержка такого режима была одой из главных функций сети. Устройства PAD сетей X.25 существовали именно для того, чтобы обеспечить удаленный доступ к мэйнфреймам для пользователей, находившихся в других городах и работавших за простыми алфавитно-цифровыми терминалами.

Режим удаленного управления обеспечивается специальным протоколом прикладного уровня, работающим поверх протоколов, реализующих транспортное соединение удаленного узла с компьютерной сетью. Существует большое количество протоколов удаленного управления, как стандартных, так и фирменных. Для IP-сетей наиболее старым протоколом этого типа является telnet (RFC 854).

Протокол **telnet**, который работает в архитектуре «клиент-сервер», обеспечивает эмуляцию алфавитно-цифрового терминала, ограничивая пользователя режимом командной строки.

При нажатии клавиши соответствующий код перехватывается клиентом telnet, помещается в TCP-сообщение и отправляется через сеть узлу, которым пользователь хочет управлять. При поступлении на узел назначения код нажатой клавиши извлекается из TCP-сообщения сервером telnet и передается операционной системе (ОС) узла. ОС рассматривает сеанс telnet как один из сеансов локального пользователя. Если ОС реагирует на нажатие клавиши выводом очередного символа на экран, то для сеанса удаленного пользователя этот символ также упаковывается в TCP-сообщение и по сети отправляется удаленному узлу. Клиент telnet извлекает символ и отображает его в окне своего терминала, эмулируя терминал удаленного узла.

Протокол telnet был реализован в среде Unix и наряду с электронной почтой и FTP-доступам к архивам файлов был популярным сервисом Интернета. Сегодня этот протокол редко используется в публичных доменах Интернета, так как никто не хочет предоставлять посторонним лицам возможность управлять собственным компьютером. Хотя для защиты от несанкционированного доступа в технологии telnet применяются пароли, они передаются через сеть в виде обычного текста, поэтому могут быть легко перехвачены и использованы. Поэтому telnet применяется преимущественно в пределах одной локальной сети, где возможностей для перехвата пароля гораздо меньше. Сегодня основной областью применения telnet является управление не компьютерами, а коммуникационными устройствами: маршрутизаторами, коммутаторами и хабами. Таким образом, он уже скорее не пользовательский протокол, а протокол администрирования, то есть альтернатива SNMP.

Тем не менее отличие между протоколами telnet и SNMP принципиальное. Telnet предусматривает обязательное участие человека в процессе администрирования, так как, по сути, он только транслирует команды, которые вводит администратор при конфигурировании или мониторинге маршрутизатора или другого коммуникационного устройства. Протокол SNMP наоборот рассчитан на автоматические процедуры мониторинга и управления, хотя и не исключает возможности участия администратора в этом процессе. Для устранения опасности, порождаемой передачей паролей в открытом виде через сеть, коммуникационные устройства усиливают степень своей защиты. Обычно применяется многоуровневая схема доступа, когда открытый пароль дает возможность только чтения базовых характеристик конфигурации устройства, а доступ к средствам изменения конфигурации требует другого пароля, который уже не передается в открытом виде.

Удаленное управление также возможно и в графическом режиме. Для Unix стандартом де-факто является система X Window, являющаяся разработкой Массачусетсского технологического института (Massachusetts Institute of Technology, MIT). Для Windows существует ряд фирменных протоколов управления, например VNC (http://www.realvnc.com); свободно распространяемая реализация этого протокола существует и для Unix.

Удаленное управление имеет свои достоинства и недостатки. Для пользователя часто удобно задействовать более мощный компьютер, установленный в сети предприятия, а не свой домашний. Кроме того, получив терминальный доступ, он может запустить на удаленном компьютере любое приложение, а не только сервис WWW или FTP. Еще одно преимущество заключается в том, что пользователь фактически получает все права пользователя внутренней сети предприятия, в то время как в режиме удаленного узла его права обычно ограничены администратором.

Удаленное управление также очень экономично потребляет пропускную способность сети, особенно при эмуляции режима командной строки. Действительно, в этом случае по сети передаются только коды клавиш и экранные символы, а не файлы или страницы веб-документов.

Недостаток удаленного управления состоит в его опасности для сети предприятия при несанкционированном доступе. Кроме того, администратору трудно контролировать потребление ресурсов компьютера, находящегося под удаленным управлением.

Коммутируемый аналоговый доступ

Основная идея коммутируемого доступа состоит в том, чтобы задействовать имеющуюся сеть PSTN для коммутируемого соединения между компьютером домашнего пользователя

и сервером удаленного доступа, установленным на границе телефонной и компьютерной сетей. Компьютер пользователя подключается к телефонной сети с помощью коммутируемого модема, который поддерживает стандартные процедуры набора номера и имитирует работу телефонного аппарата для установления соединения с RAS. Коммутируемый доступ может быть аналоговым или цифровым, в зависимости от типа абонентского окончания сети. В этом разделе мы рассмотрим доступ через аналоговые окончания, а в следующем — через цифровые.

Принцип работы телефонной сети

Первые телефонные сети были полностью аналоговыми, так как в них абонентское устройство (телефонный аппарат) преобразовывало звуковые колебания, являющиеся аналоговыми сигналами, в колебания электрического тока (также аналоговые сигналы). Коммутаторы телефонной сети тоже передавали пользовательскую информацию в аналоговой форме, перенося эти сигналы в другую область частотного спектра с помощью методов частотного уплотнения (FDM), описанных в главе 9.

Сегодня в телефонных сетях голос между коммутаторами все чаще передается в цифровой форме по каналам PDH/SDH с помощью технологии TDM. Однако абонентские окончания остаются в основном аналоговыми, что позволяет пользоваться теми же сравнительно простыми и недорогими аналоговыми телефонными аппаратами, что и раньше.

Типичная структура телефонной сети представлена на рис. 22.4. Сеть образована некоторым количеством телефонных коммутаторов, которые соединены между собой цифровыми или, в редких случаях, аналоговыми каналами. Топология связей между телефонными коммутаторами в общем случае носит произвольный характер, хотя часто имеет место многоуровневая иерархия, когда несколько коммутаторов нижнего уровня подключаются к коммутатору более высокого уровня и т. п.

К коммутаторам нижнего уровня с помощью абонентских окончаний, которые представляют собой медные пары, подключаются телефонные аппараты абонентов. Обычно длина абонентского окончания не превышает одного-двух километров, однако иногда оператор вынужден использовать и более протяженные окончания, до 5–6 км, если имеется несколько удаленных абонентов, для которых строительство отдельной точки присутствия экономически неоправданно.

Телефонная сеть, как и любая сеть с коммутацией каналов, требует обязательной процедуры предварительного установления соединения между абонентскими устройствами. В случае успеха этой процедуры в сети устанавливается канал между абонентами, через который они могут вести разговор. Процедура установления соединения реализуется с помощью *сигнального протокола*. Напомним, что в аналоговых телефонных сетях каждому абонентскому соединению выделяется полоса пропускания шириной в 4 кГц. Из этой полосы 3,1 кГц предназначается для передачи собственно голоса, а оставшиеся 900 Гц служат для передачи сигнальной информации между аналоговыми коммутаторами, а также в качестве защитной полосы частот между каналами, выделенными различным пользователям.

Существует большое количество различных сигнальных протоколов, разработанных за долгие годы существования телефонных сетей. Они делятся на два класса: сигнальные протоколы UNI работают между телефоном пользователя и первым коммутатором сети, а сигнальные протоколы NNI — между коммутаторами сети. Так как модем подключается к телефонной сети в качестве абонентского устройства, то он должен поддерживать только протокол UNI.

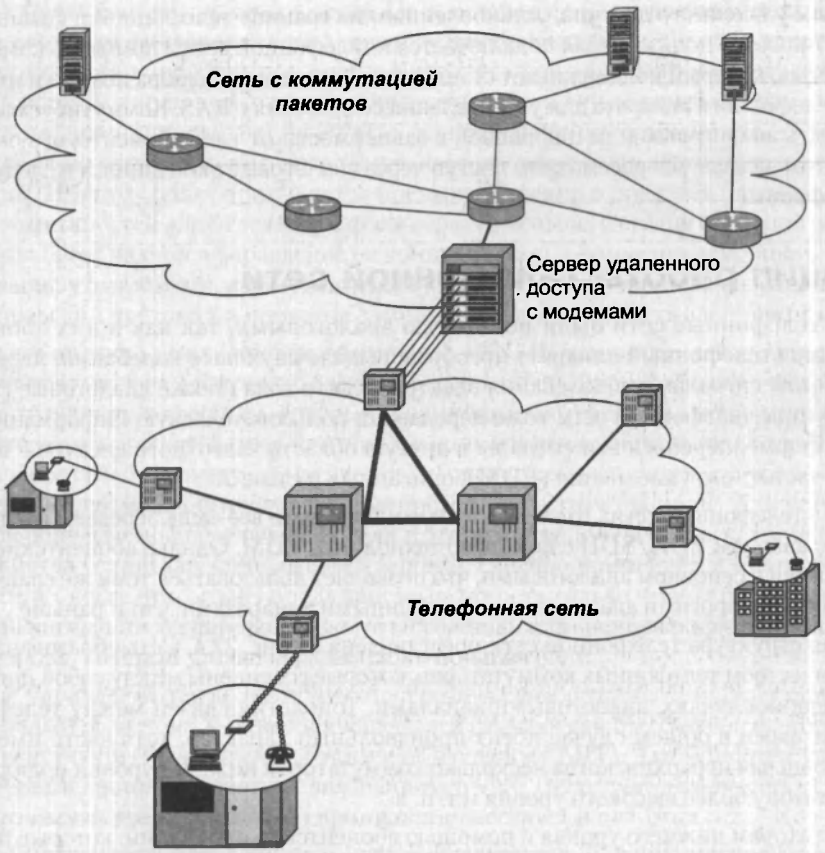


Рис. 22.4. Доступ через телефонную сеть с аналоговыми окончаниями

Аналоговый телефон — это достаточно примитивное устройство, поэтому поддерживаемый им сигнальный протокол также предельно прост. Процедура вызова абонента обычно представляет собой последовательность замыканий и размыканий электрической цепи, образуемой проводами абонентского окончания. В ответ на первое замыкание телефонный коммутатор подает на абонентскую цепь некоторое напряжение, которое воспроизводится в виде постоянного гудка динамика телефонной трубки. Человек активно участвует в процедуре вызова, набирая в ответ на гудок цифры вызываемого номера.

Существует два способа передачи номера в сеть. При **импульсном наборе** каждая цифра передается соответствующим числом последовательных импульсов размыкания-замыкания частотой 10 или 20 Гц.

При **тоновом наборе** (Dual Tone Multi Frequency, DTMF) для кодирования цифр и символов используется комбинация сигналов двух групп: низкочастотной (697, 770, 852 и 941 Гц) и высокочастотной (1209, 1336, 1477 и 1633 Гц).

Сочетания этих частот дают 16 комбинаций кодирования, как показано в табл. 22.1.

Частота 1633 Гц является расширением стандарта DTMF, с помощью которого кодируются дополнительные символы A, B, C и D, отсутствующие на стандартной клавиатуре телефонов, но используемые модемами и некоторыми приложениями.

Таблица 22.1. Кодирование цифр и символов при тоновом наборе

1209 Гц	1336 Гц	1477 Гц	1633 Гц	
1	2	3	A	697 Гц
4	5	6	B	770 Гц
7	8	9	C	852 Гц
*	0	#	D	941 Гц

Тоновый набор выполняется с частотой 10 Гц сигналами длительностью в 50 мс с паузами также в 50 мс.

Так как одна цифра номера при импульсном наборе передается несколькими импульсами, а при тоновом наборе — одним сигналом, то скорость тонового набора в несколько раз выше, чем импульсного.

После приема такого условного «сообщения» от телефонного аппарата первый коммутатор телефонной сети маршрутизирует сообщение дальше. Если этот коммутатор является цифровым, то он преобразует поступающий от абонента аналоговый сигнал в цифровую форму.

Чтобы добиться развитой логики обработки вызовов, современные телефонные коммутаторы используют протоколы **сигнальной системы 7** (Signaling System 7, SS7), в которых применяется техника коммутации пакетов. Эти протоколы построены в соответствии с моделью OSI, покрывая уровни от физического до прикладного. И хотя мы еще не раз будем упоминать SS7, подробное рассмотрение этих протоколов выходит за рамки темы данной книги, их описание можно найти в учебниках, посвященных телефонии.

Нужно подчеркнуть, что пользовательские данные по-прежнему передаются в телефонных сетях с помощью техники коммутации каналов, а техника коммутации пакетов требуется сигнальным протоколам только для установления соединения. Наряду с протоколами SS7 в телефонной сети может задействоваться большое количество более старых сигнальных протоколов, в том числе аналоговых.

Удаленный доступ через телефонную сеть

Для того чтобы получить доступ в Интернет или корпоративную сеть через телефонную сеть, модем пользователя должен выполнить вызов по одному из номеров, присвоенному модемам, находящимся на сервере удаленного доступа. После установления соединения между модемами в телефонной сети образуется канал с полосой пропускания около 4 кГц. Точное значение ширины имеющейся в распоряжении модемов полосы зависит от типа телефонных коммутаторов на пути от модема пользователя до модема RAS и от поддерживаемых ими сигнальных протоколов. В любом случае, эта полоса не превышает 4 кГц, что принципиально ограничивает скорость передачи данных модемом.

Наивысшим достижением современных модемов на канале тональной частоты является скорость в 33,6 Кбит/с, если на пути следования информации приходилось выполнять *аналого-цифровое преобразование*, и 56 Кбит/с, если преобразование было *цифро-аналоговым*. Такая асимметрия связана с тем, что аналого-цифровое преобразование вносит существенно более значительные искажения в передаваемые дискретные данные, чем цифро-аналоговое.

Очевидно, что такие скорости нельзя назвать приемлемыми для большинства современных приложений, которые широко используют графику и другие мультимедийные формы представления данных.

Модемы RAS обычно устанавливаются в точке присутствия поставщика услуг, при этом, естественно, совсем не обязательно, чтобы это был тот же самый поставщик услуг, который предоставляет доступ данному удаленному пользователю. В 80-е годы и в первой половине 90-х, когда Интернет еще не был столь популярен, многие крупные корпорации самостоятельно предоставляли удаленный доступ своим сотрудникам. В этом случае сервер удаленного доступа устанавливался в ближайшей к локальной сети штаб-квартиры корпорации точке присутствия или же в помещении самой штаб-квартиры. Сотрудники корпорации, работающие дома или находящиеся в командировке, присоединяли свои модемы к локальному поставщику услуг и звонили на модем сервера удаленного доступа корпорации. Иногда это был и международный звонок, если сотрудник находился в командировке в другой стране. Компьютерный трафик проходил основную часть пути по телефонной сети, и стоимость такого доступа зависела от расстояния, что характерно для телефонных сетей.

Сегодня Интернет позволяет использовать телефонную сеть гораздо экономичнее. Она нужна теперь не для соединения с RAS предприятия, а для соединения с RAS поставщика услуг Интернета. Если же целью пользователя является доступ не в Интернет, а в корпоративную сеть, то он задействует Интернет как промежуточную сеть, которая ведет к корпоративной сети (также подключенной к Интернету). Поскольку плата за доступ в Интернет не зависит от расстояния до узла назначения, удаленный доступ к ресурсам корпорации стал сегодня намного дешевле даже с учетом оплаты за локальный телефонный звонок и доступ в Интернет. Правда, при такой двухступенчатой схеме доступа пользователю приходится выполнять аутентификацию дважды — при доступе к RAS поставщика услуг и при доступе к RAS предприятия. Существуют протоколы, которые исключают подобное дублирование, например **двухточечный протокол туннелирования** (Point-to-Point Tunneling Protocol, PPTP). При работе PPTP сервер удаленного доступа поставщика услуг передает транзитом запрос пользователя серверу аутентификации предприятия и, в случае положительного ответа соединяет пользователя через Интернет с корпоративной сетью.

RAS может подключаться к телефонному коммутатору с помощью как аналоговых, так и цифровых окончаний. Мощные серверы удаленного доступа, оснащенные несколькими десятками модемов, обычно подключаются с помощью цифровых окончаний через линии связи T1/E1. В этом случае при передаче информации из сети передачи данных к пользователю аналого-цифровое преобразование не выполняется, поэтому скорость передачи данных в этом направлении (нисходящем) может достигать 56 Кбит/с. Однако это возможно только в том случае, когда все телефонные коммутаторы вдоль пути к пользователю являются цифровыми. В том же случае, когда хотя бы один телефонный коммутатор является аналоговым, максимальная скорость обмена в нисходящем направлении, как и в исходящем (в направлении от пользователя к сети), ограничивается значением 33,6 Кбит/с.

Модемы

Хотя коммутируемый модем предоставляет компьютеру услуги физического уровня, сам он представляет собой устройство, в котором реализованы функции двух нижних уровней модели OSI: физического и канального. Канальный уровень нужен модему для того, чтобы

выявлять и исправлять ошибки, появляющиеся из-за искажений битов при передаче через телефонную сеть. Вероятность битовой ошибки в этом случае довольно высока, поэтому функция исправления ошибок является очень важной для модема. Для протокола, который работает поверх модемного соединения между удаленным компьютером и RAS, канальный протокол модема прозрачен, его работа проявляется только в том, что интенсивность битовых ошибок (BER) снижается до приемлемого уровня. Так как в качестве канального протокола между компьютером и RAS сегодня в основном используется протокол PPP, который не занимается восстановлением искаженных и потерянных кадров, способность модема исправлять ошибки оказывается весьма полезной.

Протоколы и стандарты модемов определены в рекомендациях ITU-T серии V и делятся на три группы:

- стандарты, определяющие скорость передачи данных и метод кодирования;
- стандарты исправления ошибок;
- стандарты сжатия данных.

Стандарты метода кодирования и скорости передачи данных. Модемы являются одними из наиболее старых и заслуженных устройств передачи данных; в процессе своего развития они прошли долгий путь, прежде чем научились работать на скоростях до 56 Кбит/с.

Первые модемы работали со скоростью 300 бит/с и исправлять ошибки не умели. Эти модемы функционировали в асинхронном режиме, означающем, что каждый байт передаваемой компьютером информации передавался асинхронно по отношению к другим байтам, для чего он сопровождался стартовыми и стоповыми символами, отличающимися от символов данных. Асинхронный режим упрощает устройство модема и повышает надежность передачи данных, но существенно снижает скорость передачи, так как каждый байт дополняется одним или двумя избыточными старт-стопными символами.

Современные модемы могут работать как в асинхронном, так и синхронном режимах.

Переломным моментом в истории развития модемов стало принятие **стандарта V.34**, который повысил максимальную скорость передачи данных в два раза, с 14 до 28 Кбит/с по сравнению со своим предшественником — стандартом V.32. Особенностью стандарта V.34 являются *процедуры динамической адаптации* к изменениям характеристик канала во время обмена информацией. В V.34 определено 10 согласительных процедур, по которым модемы после тестирования линии выбирают свои основные параметры: несущую полосу и полосу пропускания, фильтры передатчика и др. Адаптация осуществляется в ходе сеанса связи без прекращения и без разрыва установленного соединения. Возможность такого адаптивного поведения была обусловлена развитием техники интегральных схем и микропроцессоров. Первоначальное соединение модемов проводится по стандарту V.21 на минимальной скорости 300 бит/с, что позволяет работать на самых плохих линиях. Затем модемы продолжают переговорный процесс до тех пор, пока не достигают максимально возможной в данных условиях производительности. Применение адаптивных процедур сразу позволило поднять скорость передачи данных более чем в 2 раза по сравнению с предыдущим стандартом — V.32 bis.

Принципы адаптивной настройки к параметрам линии были развиты в **стандарте V.34+**. Стандарт V.34+ позволил несколько повысить скорость передачи данных за счет усовершенствования метода кодирования. Один передаваемый кодовый символ несет в новом стандарте в среднем не 8,4 бита, как в протоколе V.34, а 9,8. При максимальной скорости

передачи кодовых символов в 3429 бод (это ограничение преодолеть нельзя, так как оно определяется полосой пропускания канала тональной частоты) усовершенствованный метод кодирования дает скорость передачи данных в 33,6 Кбит/с (3429 × 9,8 = 33 604).

Протоколы V.34 и V.34+ позволяют работать на 2-проводной выделенной линии в дуплексном режиме. Дуплексный режим передачи в стандартах V.34, V.34+ поддерживается не частотным разделением канала, а одновременной передачей данных в обоих направлениях. Принимаемый сигнал определяется вычитанием с помощью процессоров DSP передаваемого сигнала из общего сигнала в канале. Для этой операции используются также процедуры эхо-подавления, так как передаваемый сигнал, отражаясь от ближнего и дальнего концов канала, вносит искажения в общий сигнал.

ПРИМЕЧАНИЕ

Заметьте, что метод передачи данных, описанный в проекте стандарта 802.3ab, определяющего работу технологии Gigabit Ethernet на витой паре категории 5, взял многое из стандартов V.32–V.34+.

Стандарт V.90 описывает технологию недорогого и быстрого доступа пользователей к сетям поставщиков услуг. Этот стандарт предлагает асимметричный обмен данными: со скоростью до 56 Кбит/с из сети и со скоростью до 33,6 Кбит/с в сеть. Стандарт совместим со стандартом V.34+. Именно этот стандарт имелся в виду в предыдущем разделе, когда мы говорили о возможности нисходящей передачи данных со скоростью 56 Кбит/с при условии, что вдоль всего пути не встретится ни одного аналого-цифрового преобразователя.

В **стандарте V.92** учитывается возможность принятия модемом второго вызова во время соединения. В таких случаях современные станции передают на телефонный аппарат специальные двойные тоновые сигналы, так что абонент может распознать эту ситуацию и, нажав на аппарате кнопку Flash, переключиться на второе соединение, переведя первое соединение в режим удержания. Модемы предыдущих стандартов в таких случаях просто разрывают соединение, что не всегда удобно для абонента — может быть в этот момент он заканчивает загружать из Интернета большой файл и вся его работа пропадает.

Типовая структура соединения двух компьютеров или локальных сетей через маршрутизатор с помощью аналоговых окончаний приведена на рис. 22.5.

Рис. 22.5. Соединение компьютеров с помощью коммутируемых модемов

Коррекция ошибок. Для модемов, работающих с DTE по асинхронному интерфейсу, комитет CCITT разработал **протокол коррекции ошибок V.**42. До его принятия в модемах, работающих по асинхронному интерфейсу, коррекция ошибок обычно выполнялась

по фирменным протоколам Microcom. Эта компания реализовала в своих модемах несколько разных процедур коррекции ошибок, назвав их сетевыми протоколами Microcom (Microcom Networking Protocol, MNP) классов 2–4.

В стандарте V.42 основным является другой протокол — **протокол доступа к линии связи для модемов** (Link Access Protocol for Modems, LAP-M). Однако стандарт V.42 поддерживает и процедуры MNP 2–4, поэтому модемы, соответствующие рекомендации V.42, позволяют устанавливать связь без ошибок с любым модемом, поддерживающим этот стандарт, а также с любым MNP-совместимым модемом. Протокол LAP-M принадлежит описанному в главе 22 семейству HDLC и в основном работает так же, как и другие протоколы этого семейства — с установлением соединения, кадрированием данных, нумерацией кадров и восстановлением кадров с поддержкой метода скользящего окна. Основное отличие от других протоколов этого семейства — более развитые переговорные процедуры, для которых в протоколе LAP-M предусмотрены дополнительные типы кадров — XID и BREAK.

С помощью кадров взаимной идентификации (Exchange Identification, XID) модемы при установлении соединения могут договориться о некоторых параметрах протокола, например о максимальном размере поля данных кадра, величине тайм-аута при ожидании квитанции, размере окна и т. п. Эта процедура напоминает переговорные процедуры протокола PPP. Команда *BREAK* служит для уведомления модема-напарника о том, что поток данных временно приостанавливается. При асинхронном интерфейсе с DTE такая ситуация может возникнуть. Команда *BREAK* посылается в ненумерованном кадре и не влияет на нумерацию потока кадров сеанса связи. После возобновления поступления данных модем продолжает работать так, как если бы паузы в передаче не было.

Сжатие данных. Почти все современные модемы при работе по асинхронному интерфейсу поддерживают **стандарты сжатия данных CCITT V.42bis** и **MNP-5** (обычно с коэффициентом 1:4, некоторые модели — до 1:8). Сжатие данных увеличивает пропускную способность линии связи. Передающий модем автоматически сжимает данные, а принимающий их восстанавливает. Модем, поддерживающий протокол сжатия, всегда пытается установить связь со сжатием данных, но если второй модем этот протокол не поддерживает, то и первый модем переходит на обычную связь без сжатия.

При работе модемов по синхронному интерфейсу наиболее популярным является протокол **сжатия синхронных потоков данных** (Synchronous Data Compression, SDC) компании Motorola.

Коммутируемый доступ через сеть ISDN

Назначение и структура ISDN

Целью создания технологии ISDN (Integrated Services Digital Network — **цифровая сеть с интегрированным обслуживанием**) было построение всемирной сети, которая должна была прийти на смену телефонной сети и, будучи такой же доступной и распространенной, предоставлять миллионам своих пользователей разнообразные услуги, как телефонные, так и передачи данных. Передача телевизионных программ по ISDN не предполагалась, поэтому было решено ограничиться пропускной способностью абонентского окончания для массовых пользователей в 128 Кбит/с.

Если бы цель разработчиков ISDN была достигнута в полной мере, то проблема доступа домашних пользователей к Интернету и корпоративным сетям была бы окончательно решена. Однако по многим причинам внедрение ISDN происходило очень медленно — процесс, который начался в 80-е годы, растянулся больше чем на десять лет, так что к моменту появления в домах пользователей некоторые услуги ISDN просто морально устарели. Так, скорость доступа 128 Кбит/с сегодня уже достаточна не для всех пользователей. Существует, правда, такой интерфейс ISDN, который обеспечивает скорость доступа до 2 Мбит/с, но он достаточно дорог для массового пользователя и его обычно применяют только предприятия для подключения своих сетей.

Хотя сеть ISDN и не стала той новой публичной сетью, на роль которой она претендовала, ее услуги сегодня достаточно доступны. Далее мы рассмотрим структуру этой сети и ее возможности в отношении организации удаленного доступа.

Архитектура сети ISDN предусматривает несколько видов услуг (рис. 22.6):

- некоммутируемые средства (выделенные цифровые каналы);
- коммутируемая телефонная сеть общего пользования;
- сеть передачи данных с коммутацией каналов;
- сеть передачи данных с коммутацией пакетов;
- сеть передачи данных с трансляцией кадров (режим сети Frame Relay);
- средства контроля и управления работой сети.

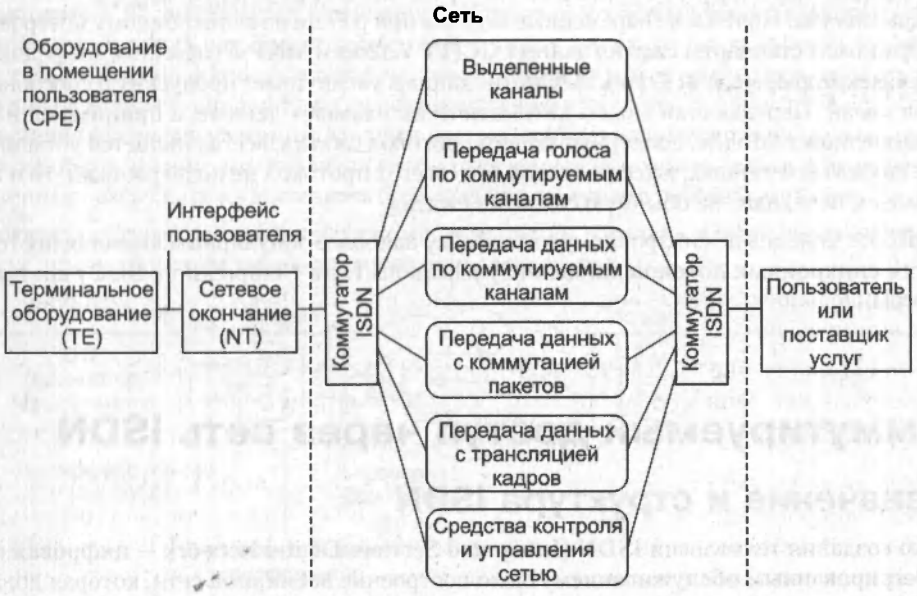


Рис. 22.6. Услуги сети ISDN

Как видно из приведенного списка, транспортные службы сетей ISDN действительно покрывают очень широкий спектр услуг, включая популярные услуги сети Frame Relay. Стандарты ISDN описывают также ряд услуг прикладного уровня: факсимильную связь

на скорости 64 Кбит/с, телексную связь на скорости 9600 бит/с, видеотекс на скорости 9600 бит/с и некоторые другие.

Все услуги основаны на передаче информации в цифровой форме. Пользовательский интерфейс также является цифровым, то есть все его абонентские устройства (телефон, компьютер, факс) должны передавать в сеть цифровые данные. Организация **цифрового абонентского окончания** (Digital Subscriber Line, DSL) стала одним из серьезных препятствий на пути распространения ISDN, так как требовала модернизации миллионов абонентских окончаний.

На практике не все сети ISDN поддерживают все стандартные службы. Служба Frame Relay, хотя и была разработана в рамках сети ISDN, реализуется, как правило, с помощью отдельной сети коммутаторов кадров, не пересекающейся с сетью коммутаторов ISDN.

Базовой скоростью сети ISDN является скорость канала DS-0, то есть 64 Кбит/с. Эта скорость ориентируется на самый простой метод кодирования голоса — PCM, хотя дифференциальное кодирование и позволяет передавать голос с тем же качеством на скорости 32 или 16 Кбит/с.

Одной из оригинальных идей, положенных в основу ISDN, является совместное использование принципов коммутации каналов и пакетов. Однако сеть с коммутацией пакетов, работающая в составе ISDN, выполняет только служебные функции — с ее помощью передаются сообщения сигнального протокола. А вот основная информация, то есть сам голос, по-прежнему передается через сеть с коммутацией каналов. В таком разделении функций есть вполне понятная логика — сообщения о вызове абонентов образуют пульсирующий трафик, поэтому его эффективнее передавать по сети с коммутацией пакетов.

Интерфейсы BRI и PRI

Одним из основных принципов ISDN является предоставление пользователю стандартного интерфейса, с помощью которого пользователь может запрашивать у сети разнообразные услуги. Этот интерфейс образуется между двумя типами оборудования, устанавливаемого в помещении пользователя (Customer Premises Equipment, CPE). К этому оборудованию относится:

- **терминальное оборудование** (Terminal Equipment, TE) пользователя (компьютер с соответствующим адаптером, маршрутизатор, телефонный аппарат);
- **сетевое окончание** (Network Termination, NT), которое представляет собой устройство, завершающее линию связи с ближайшим коммутатором ISDN.

Пользовательский интерфейс основан на каналах трех типов: B, D и H.

Каналы типа B обеспечивают передачу пользовательских данных (оцифрованного голоса, компьютерных данных или смеси голоса и данных) с более низкими скоростями, чем 64 Кбит/с. Разделение данных выполняется с помощью техники TDM. Разделением канала B на подканалы в этом случае должно заниматься пользовательское оборудование, сеть ISDN всегда коммутирует целые каналы типа B. Каналы типа B могут соединять пользователей с помощью техники коммутации каналов друг с другом, а также образовывать так называемые полупостоянные соединения, которые эквиваленты соединениям выделенных каналов обычной телефонной сети. Канал типа B может также подключать пользователя к коммутатору сети X.25.

Канал типа D является каналом доступа к служебной сети с коммутацией пакетов, передающей сигнальную информацию со скоростью 16 или 64 Кбит/с. Передача адресной информации, на основе которой осуществляется коммутация каналов типа B в коммутаторах сети, является основной функцией канала D. Другой его функцией является поддержание сервиса низкоскоростной сети с коммутацией пакетов для пользовательских данных. Обычно этот сервис выполняется сетью в то время, когда каналы типа D свободны от выполнения основной функции.

Каналы типа H предоставляют пользователям возможности высокоскоростной передачи данных со скоростью 384 Кбит/с (H0), 1536 Кбит/с (H11) или 1920 Кбит/с (H12). На них могут работать службы высокоскоростной передачи факсов, видеоинформации, качественного воспроизведения звука.

Пользовательский интерфейс ISDN представляет собой набор каналов определенного типа и с определенными скоростями. Сеть ISDN поддерживает два вида пользовательского интерфейса с начальной (Basic Rate Interface, BRI) и основной (Primay Rate Interface, PRI) скоростями передачи данных.

Начальный интерфейс ISDN предоставляет пользователю два канала по 64 Кбит/с для передачи данных (каналы типа B) и один канал с пропускной способностью 16 Кбит/с для передачи управляющей информации (канал типа D). Все каналы работают в дуплексном режиме. В результате суммарная скорость интерфейса BRI для пользовательских данных составляет 144 Кбит/с по каждому направлению, а с учетом служебной информации — 192 Кбит/с. Различные каналы пользовательского интерфейса разделяют один и тот же физический двухпроводный кабель по технологии TDM, то есть являются логическими, а не физическими каналами. Данные по интерфейсу BRI передаются кадрами, состоящими из 48 бит. Каждый кадр содержит по 2 байта каждого из двух каналов B, а также 4 бита канала D. Передача кадра длится 250 мс, что обеспечивает скорость передачи данных 64 Кбит/с для каналов B и 16 Кбит/с — для канала D. Помимо битов данных кадр содержит служебные биты для синхронизации кадров, а также обеспечения нулевой постоянной составляющей электрического сигнала. Интерфейс BRI может поддерживать не только схему 2B + D, но и B + D и просто D.

Начальный интерфейс стандартизован в рекомендации I.430.

Основной интерфейс ISDN предназначен для пользователей с повышенными требованиями к пропускной способности сети. Интерфейс PRI поддерживает либо схему 30B + D, либо схему 23B + D. В обеих схемах канал D обеспечивает скорость 64 Кбит/с. Первый вариант предназначен для Европы, второй — для Северной Америки и Японии. Ввиду большой популярности скорости цифровых каналов 2,048 Мбит/с в Европе и скорости 1,544 Мбит/с в остальных регионах привести стандарт на интерфейс PRI к общему варианту не удалось.

Возможны варианты интерфейса PRI с меньшим количеством каналов типа B, например 20B + D. Каналы типа B могут объединяться в один логический высокоскоростной канал с общей скоростью до 1920 Кбит/с. При установке у пользователя нескольких интерфейсов PRI все они могут иметь один канал типа D, при этом количество каналов B в том интерфейсе, который не имеет канала D, может увеличиваться до 24 или 31.

Основной интерфейс может быть также основан на каналах типа H. При этом общая пропускная способность интерфейса все равно не должна превышать 2,048 или 1,544 Мбит/с. Для каналов H0 возможны интерфейсы 3H0 + D для американского варианта и 5H0 + D для европейского. Для каналов H1 возможен интерфейс, состоящий только из одно-

го канала H11 (1,536 Мбит/с) для американского варианта или одного канала H12 (1,920 Мбит/с) и одного канала D для европейского варианта. Кадры интерфейса PRI имеют структуру кадров DS-1 для каналов T1 или E1.

Основной интерфейс PRI стандартизован в рекомендации I.431.

ВНИМАНИЕ

Как каналы B, так и каналы D являются логическими каналами абонентского окончания, которое физически представляет собой одну витую пару. Каналы D и B образуются путем применения техники TDM к физической среде, образуемой этой витой парой.

Стек протоколов ISDN

В сети ISDN существует два стека протоколов: стек каналов типа D и стек каналов типа B (рис. 22.7).

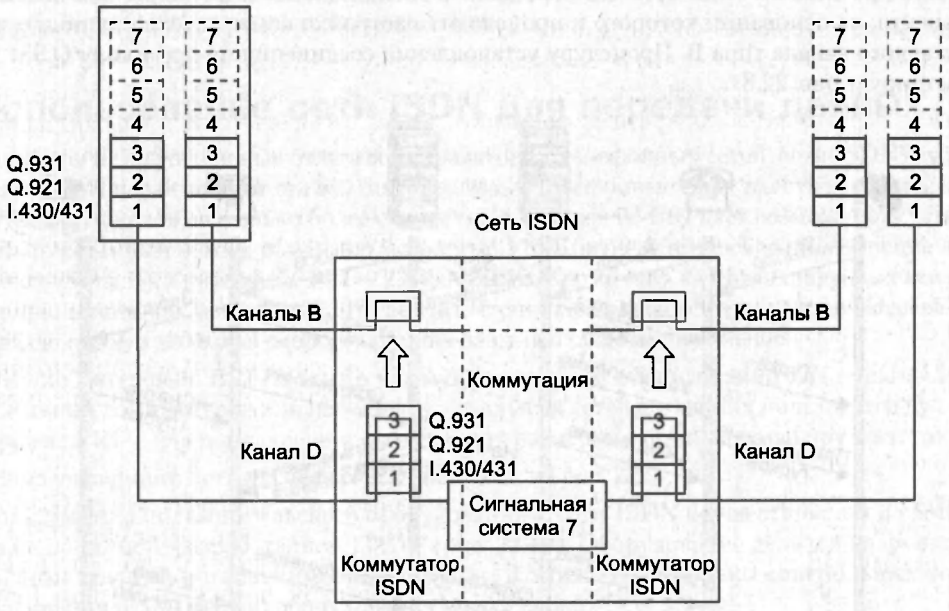


Рис. 22.7. Структура сети ISDN

Сеть каналов типа D внутри сети ISDN служит транспортной системой с коммутацией пакетов, применяемой для передачи сообщений сигнализации. Прообразом этой сети послужила технология сетей X.25. Для сети каналов D определены три уровня протоколов:

- физический протокол определяется стандартом I.430/431;
- канальный протокол LAP-D определяется стандартом Q.921;
- на сетевом уровне может использоваться протокол сигнализации Q.931, с помощью которого выполняется маршрутизация вызова абонента службы с коммутацией каналов.

Каналы типа B образуют сеть с коммутацией каналов, которая передает данные абонентов, то есть оцифрованный голос. В терминах модели OSI на каналах типа B в коммутаторах

сети ISDN определен только протокол физического уровня — протокол I.430/431. Коммутация каналов типа B происходит по указаниям, полученным по каналу D. Когда кадры протокола Q.931 маршрутизируются коммутатором, происходит одновременная коммутация очередной части составного канала от исходного абонента к конечному.

Протокол LAP-D принадлежит к семейству HDLC. Протокол LAP-D обладает всеми «родовыми чертами» этого семейства, но имеет и некоторые особенности. Адрес кадра LAP-D состоит из двух байтов — один байт определяет код службы, которой пересылаются вложенные в кадр пакеты, а второй требуется для адресации одного из терминалов, если у пользователя к абонентскому окончанию подключено несколько терминалов. Терминальное устройство ISDN может поддерживать разные услуги: установление соединения по протоколу Q.931, коммутация пакетов X.25, мониторинг сети и т. п. Протокол LAP-D обеспечивает два режима работы: с установлением соединения и без установления соединения. Последний режим используется, например, для мониторинга сети.

Протокол Q.931 является сигнальным протоколом ISDN для участка пользователь-сеть, то есть протоколом типа UNI. Он переносит в своих пакетах ISDN-адрес вызываемого абонента, на основании которого и происходит настройка коммутаторов на поддержку составного канала типа B. Процедуру установления соединения по протоколу Q.931 иллюстрирует рис. 22.8.

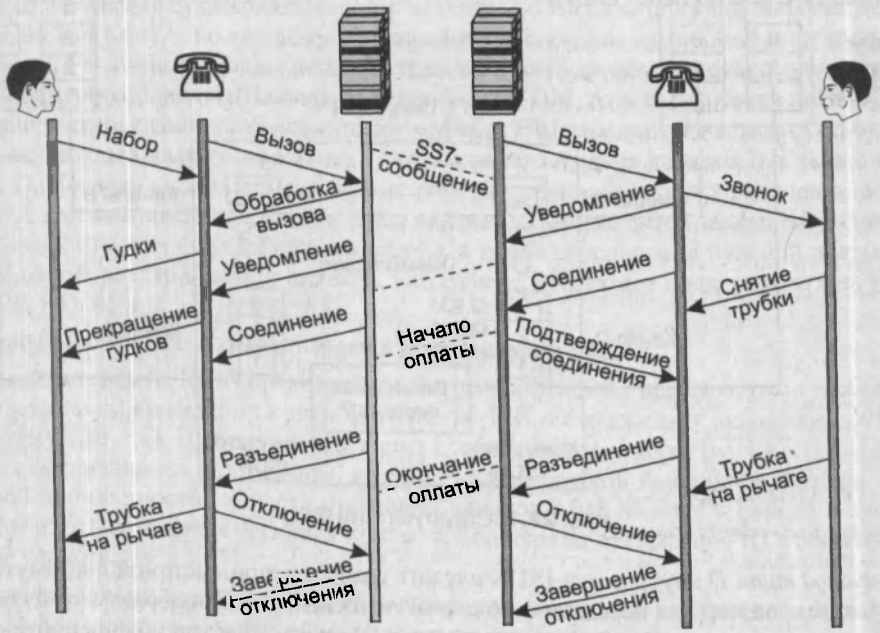


Рис. 22.8. Базовая процедура установления соединения в ISDN по протоколу Q.931

После того как пользователь снял трубку и набрал номер вызываемого абонента, телефонный аппарат ISDN формирует пакет вызова (set up) и отправляет его по каналу D коммутатору ISDN, к которому он подключен. Этот коммутатор отвечает аппарату абонента пакетом обработки вызова, с приходом которого аппарат начинает генерировать длинные гудки. Одновременно коммутатор запоминает факт запроса на установление соединения и передает принятое сообщение следующему коммутатору, адрес которого он находит по таблице, аналогичной таблице маршрутизации маршрутизаторов пакетных сетей. При

этом сообщение протокола Q.931 транслируется в сообщение начального адреса (Initial Address Message, IAM) протокола SS7 аналогичного назначения (на рисунке сообщения SS7 не детализированы). Проходя через сеть, сообщения SS7 переводят промежуточные коммутаторы в состояние готовности к установлению соединения. Выходной коммутатор сети, к которому подключен аппарат вызываемого абонента, преобразует сообщение начального адреса протокола SS7 в сообщение вызова протокола Q.931, на основании которого телефонный аппарат начинает звонить. Если абонент снимает трубку, то его аппарат генерирует сообщение соединения (connect), которое в обратном порядке проходит через все промежуточные коммутаторы (преобразованное, естественно, в соответствующее сообщение SS7). При обратном проходе коммутаторы устанавливают состояние соединения, коммутируя соответствующим образом каналы типа B.

Любое абонентское устройство ISDN должно поддерживать протокол Q.931, так что телефон ISDN намного сложнее своего аналогового коллеги. Как видно из рисунка, внутри сети сообщения Q.931 транслируются в сообщения протокола SS7, который является протоколом взаимодействия коммутатор-коммутатор (NNI), а затем снова преобразуются в сообщения Q.931 на абонентском окончании.

Использование сети ISDN для передачи данных

Несмотря на значительные отличия от аналоговых телефонных сетей, сети ISDN сегодня используются в основном так же, как аналоговые телефонные сети, то есть как сети с коммутацией каналов, но только более скоростные: интерфейс BRI дает возможность установить дуплексный режим обмена со скоростью 128 Кбит/с (логическое объединение двух каналов типа B), а интерфейс PRI — 2,048 Мбит/с. Кроме того, качество цифровых каналов гораздо выше, чем аналоговых. Это значит, что процент искаженных кадров оказывается гораздо ниже, а полезная скорость обмена данными существенно выше.

Обычно интерфейс BRI служит в коммуникационном оборудовании для подключения отдельных компьютеров или небольших локальных сетей домашних пользователей, а интерфейс PRI — для подключения сети средних размеров с помощью маршрутизатора.

Схема удаленного доступа через ISDN показана на рис. 22.9.

Подключение пользовательского оборудования к сети ISDN осуществляется в соответствии со схемой, разработанной ITU-T (рис. 22.10). Оборудование делится на функциональные группы, и в зависимости от группы различают несколько **контрольных точек** соединения разных групп оборудования между собой.

Терминальным оборудованием 1 (TE1) может быть цифровой телефон или факс-аппарат. *Контрольная точка S* соответствует точке подключения отдельного терминального устройства к устройству сетевого окончания (устройство типа NT1) или концентратору пользовательских интерфейсов (устройству типа NT2). TE1 по определению поддерживает один из пользовательских интерфейсов ISDN: BRI или PRI.

Если пользовательское терминальное оборудование TE1 подключено через интерфейс BRI, то цифровое абонентское окончание выполняется по 2-проводной схеме (как и обычное окончание аналоговой телефонной сети). Для кодирования данных на участке DSL до точки подключения к сети ISDN (*контрольная точка U*) в этом случае используется потенциальный код 2B1Q. Дуплексный режим DSL образован путем одновременной передачи сигналов по одной витой паре в обоих направлениях с эхо-подавлением и вычитанием своего сигнала из суммарного. Максимальная длина абонентского окончания для этого варианта составляет 5,5 км.

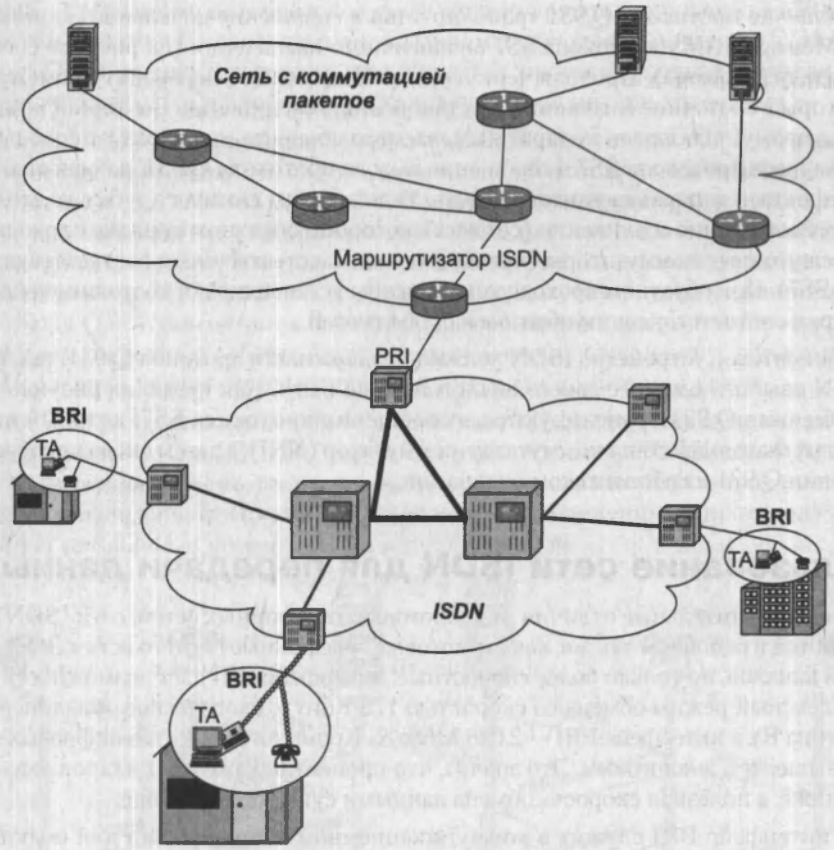


Рис. 22.9. Удаленный доступ с использованием ISDN

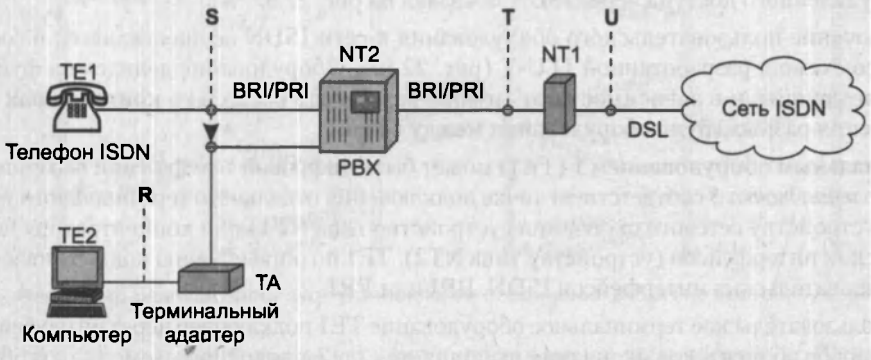


Рис. 22.10. Подключение пользовательского оборудования ISDN

При использовании терминальным оборудованием TE1 интерфейса PRI цифровое абонентское окончание должно представлять собой канал T1 или E1, то есть 4-проводную

линию с максимальной длиной около 1800 м. Соответственно на участке DSL до точки U применяется код HDB3 (Европа) или B8ZS (Америка).

Терминальное оборудование 2 (TE2) в отличие от TE1 не поддерживает интерфейсы BRI и PRI. Таким оборудованием может быть компьютер или маршрутизатор с последовательными интерфейсами, не относящимися к ISDN, например RS-232C, X.21 или V.35. Для подключения подобного оборудования к сети ISDN необходимо использовать терминальный адаптер. **Терминальный адаптер** (Terminal Adaptor, TA) согласует интерфейс TE2 с интерфейсом PRI или BRI. Для компьютеров терминальные адаптеры выпускаются в формате сетевых адаптеров. *Контрольная точка R* соответствует точке подключения терминального оборудования TE2 к TA. Тип абонентского окончания не зависит от того, работает терминальное оборудование через TA или непосредственно.

Устройства сетевого окончания 2 (NT2) представляют собой устройства канального или сетевого уровня, которые выполняют функции концентрации пользовательских интерфейсов и их мультиплексирования. Например, к этому типу оборудования относятся: офисная АТС, коммутирующая несколько интерфейсов BRI, маршрутизатор, работающий в режиме коммутации пакетов (например, по каналу D), простой мультиплексор TDM, который мультиплексирует несколько низкоскоростных каналов в один канал типа B. Точка подключения оборудования типа NT2 к абонентскому сетевому окончанию (устройству NT1) называется *контрольной точкой T*. Поскольку наличие данного типа оборудования не является обязательным (в отличие от NT1), то контрольные точки S и T объединяются и обозначаются как *контрольная точка S/T*. Физически интерфейс в точке S/T представляет собой 4-проводную линию. Для интерфейса BRI в качестве метода кодирования выбран биполярный метод AMI, причем логическая единица кодируется нулевым потенциалом, а логический ноль — чередованием потенциалов противоположной полярности. Для интерфейса PRI используются другие коды — те же, что и для интерфейсов T1 и E1, то есть соответственно B8ZS и HDB3.

Устройства сетевого окончания 1 (NT1) — это устройство физического уровня, которое согласует интерфейс BPR или PRI с цифровым абонентским окончанием (DSL), соединяющим пользовательское оборудование с сетью ISDN. Фактически NT1 представляет собой устройство типа CSU, которое согласует методы кодирования, количество используемых линий и параметры электрических сигналов. *Контрольная точка U* соответствует точке подключения устройства NT1 к сети.

ПРИМЕЧАНИЕ

Устройство NT1 может принадлежать оператору сети или пользователю (хотя всегда устанавливается в помещении пользователя). В Европе принято считать устройство NT1 частью сетевого оборудования, поэтому пользовательское оборудование (например, маршрутизатор с интерфейсом ISDN) выпускается без встроенного устройства NT1. В Северной Америке принято считать устройство NT1 принадлежностью пользовательского оборудования, поэтому пользовательское оборудование часто выпускается со встроенным устройством NT1.

Таким образом, для удаленного доступа необходимо оснастить компьютеры пользователей терминальными адаптерами, а в POP установить маршрутизатор, имеющий один или несколько интерфейсов PRI. В этом случае максимальная скорость доступа для отдельного пользователя будет равна скорости передачи двух каналов типа B, то есть 128 Кбит/с. Драйверы терминальных адаптеров ISDN умеют объединять два отдельных физических

канала типа B в один логический канал. Для этого служит расширение протокола PPP — многоканальный протокол PPP (RFC 1990).

Если пользователь удаленного доступа согласен ограничиться скоростью 64 Кбит/с, он может задействовать второй канал типа B своего интерфейса BRI для параллельной работы телефона ISDN, что невозможно сделать при применении аналогового коммутируемого модема.

Технология ADSL

Технология асимметричного цифрового абонентского окончания (Assymetric Digital Subscriber Line, ADSL) была разработана для обеспечения скоростного доступа в Интернет массовых индивидуальных пользователей, квартиры которых оснащены обычными абонентскими телефонными окончаниями. Появление технологии ADSL можно считать революционным событием для массовых пользователей Интернета, потому что для них оно означало повышение скорости доступа в десятки раз (а то и более) без какого бы то ни было изменения кабельной проводки в квартире и доме.

Для доступа через ADSL, так же как и для аналогового коммутируемого доступа, нужны телефонные абонентские окончания и модемы. Однако принципиальным отличием доступа через ADSL от коммутируемого доступа является то, что ADSL-модемы работают только в пределах абонентского окончания, в то время как коммутируемые модемы используют возможности телефонной сети, устанавливая в ней соединение «из конца в конец», которое проходит через несколько транзитных коммутаторов.

Поэтому если традиционные телефонные модемы (например, V.34, V.90) должны обеспечивать передачу данных на канале с полосой пропускания в 3100 Гц, то ADSL-модемы получают в свое распоряжение полосу порядка 1 МГц — эта величина зависит от длины кабеля, проложенного между помещением пользователя и POP, и сечения проводов этого кабеля.

Схема доступа через ADSL показана на рис. 22.11. Эта схема близка к общей схеме использования универсального абонентского окончания (см. рис. 22.2) за исключение того, что при доступе через ADSL факт наличия телевизоров у пользователей игнорируется, а доступ для телефонов и компьютеров является совместным.

ADSL-модемы, подключаемые к обоим концам короткой линии между абонентом и POP, образуют три канала: высокоскоростной нисходящий канал передачи данных из сети в компьютер, менее скоростной восходящий канал передачи данных из компьютера в сеть и канал телефонной связи, по которому передаются обычные телефонные разговоры. Передача данных в канале от сети к абоненту в стандарте ADSL 1998 года происходит со скоростью от 1,5 до 8 Мбит/с, а в канале от абонента к сети — от 16 Кбит/с до 1 Мбит/с; для телефона оставлена традиционная полоса в 4 кГц (рис. 22.12).

Для асимметрии нисходящей и восходящей скоростей полоса пропускания абонентского окончания делится между каналами также асимметрично. На рис. 22.12 показано распределение полосы между каналами, при этом приведенные значения для восходящей и нисходящей полосы являются максимальными значениями, которые модем в каждом конкретном сеансе может использовать полностью или же частично.

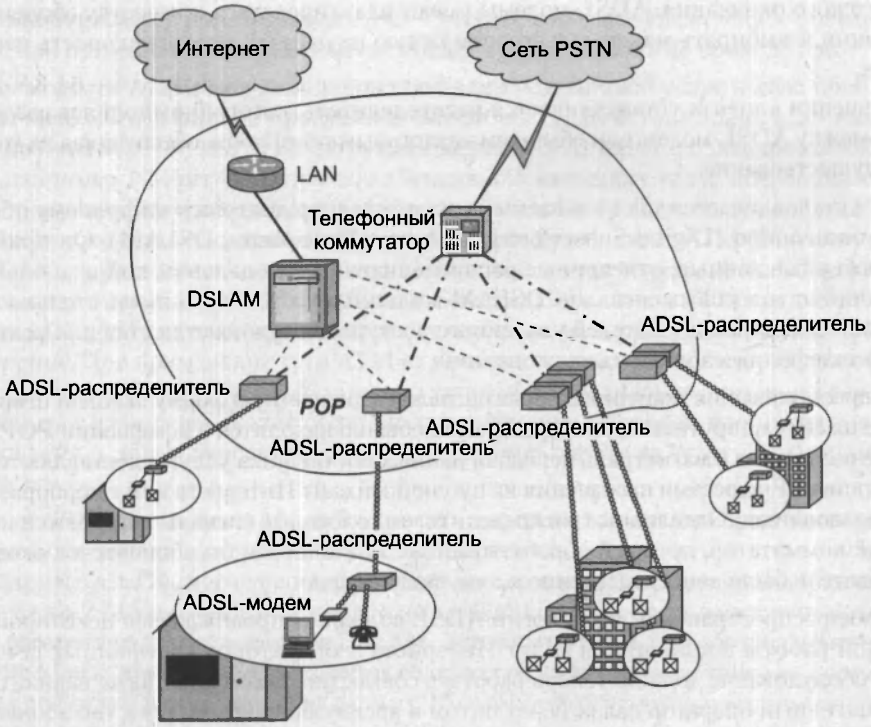


Рис. 22.11. Отличия условий работы ADSL-модемов от обычных модемов

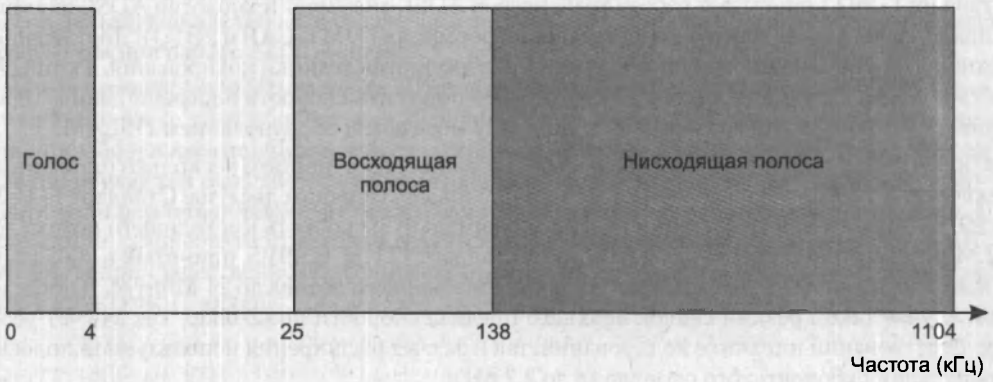


Рис. 22.12. Распределение полосы пропускания абонентского окончания между каналами ADSL

Неопределенность используемых полос частот объясняется тем, модем постоянно тестирует качество сигнала и выбирает только те части выделенного для передачи спектра, в которых соотношение сигнал/шум является приемлемым для устойчивой передачи дискретных данных. Заранее сказать, в каких частях выделенного спектра это соотношение окажется приемлемым, невозможно, так как это зависит от длины абонентского окончания, от сечения провода, от качества витой пары в целом, от помех, которые наводятся на провода

абонентского окончания. ADSL-модемы умеют адаптироваться к качеству абонентского окончания и выбирать максимально возможную на данный момент скорость передачи данных.

В помещении клиента устанавливается распределитель, который выполняет разделение частот между ADSL-модемом и обычным аналоговым телефоном, обеспечивая их совместное сосуществование.

В POP устанавливается так называемый **мультиплексор доступа к цифровому абонентскому окончанию** (Digital Subscriber Line Access Multiplexer, DSLAM). Он принимает компьютерные данные, отделенные распределителями на дальнем конце абонентских окончаний от голосовых сигналов. DSLAM-мультиплексор должен иметь столько ADSL-модемов, сколько пользователей удаленного доступа обслуживает поставщик услуг с помощью телефонных абонентских окончаний.

После преобразования модулированных сигналов в дискретную форму DSLAM отправляет данные на IP-маршрутизатор, который также обычно находится в помещении POP. Далее данные поступают в магистраль передачи данных поставщика услуг и доставляются в соответствии с IP-адресами назначения на публичный сайт Интернета или в корпоративную сеть пользователя. Отделенные распределителем голосовые сигналы передаются на телефонный коммутатор, который обрабатывает их так, как если бы абонентское окончание пользователя было непосредственно к нему подключено.

Широкое распространение технологий ADSL должно сопровождаться некоторой перестройкой работы поставщиков услуг Интернета и операторов телефонных сетей, так как их оборудование должно теперь работать совместно. Возможен также вариант, когда альтернативный оператор связи берет оптом в аренду большое количество абонентских окончаний у традиционного местного оператора или же арендует некоторое количество модемов в DSLAM.

Стандарт G.992.1 описывает работу трансиверов ADSL-модемов. Технология ADSL поддерживает несколько вариантов кодирования информации (DMT, CAP и 2B1Q). Достижения технологий xDSL во многом определяются достижениями техники кодирования, в которой за счет применения процессоров DSP удалось повысить скорость передачи данных при одновременном увеличении расстояния между модемом и оборудованием DSLAM.

За более чем десятилетнюю историю существования было принято несколько стандартов технологии ADSL, которые повысили верхний предел скорости доступа. Стандарт ITU-T G.991.2, принятый в 1999 году, повысил максимальную скорость нисходящего потока до 12 Мбит/с, а восходящего – до 1,3 Мбит/с, стандарт ITU-T G.991.5, принятый в 2003 году и известный как ADSL2+, повысил скорость нисходящего потока до 24 Мбит/с. В последнем случае такой резкий скачок верхнего предела скорости произошел как за счет усовершенствований в технике кодирования, так и за счет расширения используемой полосы пропускания абонентского окончания до 2,2 МГц.

В 2006 году был принят стандарт ITU-T G.992.3, известный под названием VDSL2 (Very high-speed DSL2 – сверхскоростное цифровое абонентское окончание 2). Этот стандарт позволяет достигать скорости нисходящего потока до 250 Мбит/с, но только на достаточно коротких расстояниях от абонента до точки присутствия оператора, в том же случае, когда это расстояние увеличивается до 1,5 км, скорость передачи данных падает до скорости стандарта ADSL2+.

Нужно подчеркнуть, что новые высокоскоростные стандарты рассчитаны в первую очередь на высококачественные телефонные абонентские окончания; в тех же случаях, когда каче-

ство проводки низкое, а расстояние до АТС — значительное, на существенное повышение скорости при применении модема нового стандарта рассчитывать не приходится.

Высокие скорости ADSL-модемов порождают для поставщиков услуг новую проблему, а именно проблему дефицита пропускной способности. Действительно, если каждый абонент доступа через ADSL будет загружать данные из Интернета с максимальной скоростью, например 1 Мбит/с, то при 100 абонентах поставщику услуг потребовался бы канал с пропускной способностью 100 Мбит/с, то есть Fast Ethernet, а если разрешить пользователям работать со скоростью 6 Мбит/с, то уже нужен канал ATM 622 Мбит/с или Gigabit Ethernet. Для обеспечения необходимой скорости многие устройства DSLAM имеют встроенный коммутатор ATM или Gigabit Ethernet. Технология ATM привлекает разработчиков DSLAM не только своей высокой скоростью, но и тем, что она ориентирована на соединение. При применении сети ATM на канальном уровне компьютер пользователя перед передачей данных должен обязательно установить соединение с сетью поставщика услуг. Это дает возможность контролировать доступ пользователей и учитывать время использования и объем переданных данных, если при оплате за услугу эти параметры учитываются.

Технология SDSL позволяет на одной паре абонентского окончания организовать два симметричных канала передачи данных. Канал тональной частоты в этом случае не предусматривается. Обычно скорости каналов в восходящем и нисходящем направлениях составляют по 2 Мбит/с, но как и у технологии ADSL, эта скорость зависит от качества линии и расстояния до оборудования DSLAM. Технология SDSL разработана в расчете на небольшие офисы, локальные сети которых содержат собственные источники информации, например веб-сайты или серверы баз данных. Поэтому характер трафика здесь ожидается скорее симметричный, так как доступ через SDSL потребуется не только к внешним сетям из локальных сетей, но и к таким источникам информации извне. В технологии SDSL используется также голосовая часть спектрального диапазона, поэтому при работе SDSL-модема нельзя параллельно с передачей данных разговаривать по обычному телефону, как это делается при работе ADSL-модема.

Широкое применение доступа через xDSL наносит еще один удар технологии ISDN. При применении этого типа абонентских окончаний пользователь получает еще и интегрированное обслуживание двух сетей: телефонной и компьютерной. Но для пользователя наличие двух сетей оказывается незаметным, для него только ясно, что он может одновременно пользоваться обычным телефоном и подключенным к Интернету компьютером. Скорость же компьютерного доступа при этом превосходит возможности интерфейса PRI сети ISDN при существенно более низкой стоимости, определяемой низкой стоимостью инфраструктуры IP-сетей.

Доступ через сети CATV

Кабельное телевидение является одной из телекоммуникационных услуг, для которой была создана собственная разветвленная инфраструктура абонентских окончаний. Хотя кабельное телевидение и уступает по распространенности телефонной сети, тем не менее количество коаксиальных абонентских окончаний, соединяющих дома и квартиры с точками присутствия поставщиков услуг, в некоторых странах стало приближаться к количеству абонентских телефонных окончаний. Учитывая, что коаксиальный кабель обладает гораздо более широкой полосой пропускания (как минимум, 700–800 МГц), абонентское окончание

CATV может вполне справиться с одновременной передачей телефонного, компьютерного и телевизионного трафиков.

Схема использования линий CATV в качестве универсальных окончаний для доступа в Интернет, телефонную сеть и сеть кабельного телевидения нами в общих чертах уже рассматривалась. Именно окончание CATV было выбрано в качестве примера на рис. 22.2. Теперь мы остановимся на некоторых деталях этого вида доступа.

Отличием абонентского окончания CATV является то, что к коаксиальному кабелю по схеме монтажного ИЛИ подключаются одновременно несколько абонентов (рис. 22.13). Это может быть несколько десятков домов или же сотен квартир многоквартирного дома. Поэтому абонентское окончание CATV представляет собой классическую разделяемую среду, которая используется, например, в сетях Ethernet на коаксиальном кабеле.

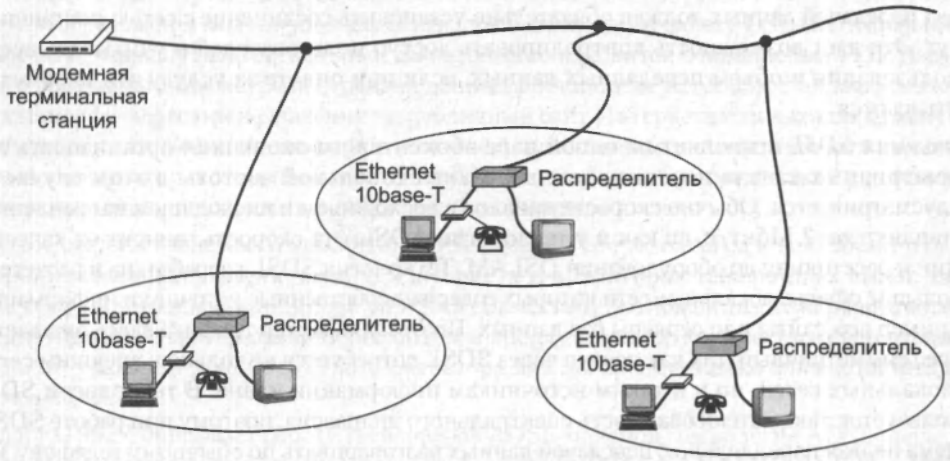


Рис. 22.13. Подключение кабельных модемов к окончанию CATV

В отсутствии кабельных модемов оборудование CATV служит для широковещательного распространения телевизионных программ до телевизионных приемников абонентов CATV из источника информации, расположенного в точке присутствия поставщика услуг. Для этого занимается диапазон частот от 50 до 550–868 МГц (точное значение зависит от национальной политики выделения частот). Каждой программе CATV выделяется в этом диапазоне полоса в 6 или 8 МГц, сигнал которой шифруется и может быть дешифрирован приемниками тех абонентов, которые подписались на прием определнной программы.

Для использования такого абонентского окончания в помещении каждого абонента высокоскоростного доступа устанавливается распределитель и кабельный модем, а в точке присутствия — головной модем, который еще называют **модемной терминальной станцией** (Cable Modem Termination Station, CMTS).

Для двунаправленной передачи компьютерных данных кабельные модемы клиентов и станция CMTS занимают неиспользуемые телевизионными программами частоты. Обычно это диапазон относительно низких частот от 5 до 50 МГц, расположенный ниже частот телевизионных программ, а также диапазон высоких частот выше 550 МГц.

Диапазон низких частот используется для менее скоростного восходящего канала, а диапазон высоких частот — для высокоскоростного нисходящего канала. Скорость передачи

данных в восходящем направлении может доходить до 10 Мбит/с, а в нисходящем — до 30–40 Мбит/с. Модемы пользователей могут взаимодействовать только со станцией CMTS.

Так как восходящий и нисходящий каналы разделены по частотам, абонентское окончание CATV образует две разделяемые среды.

Для нисходящего канала CMTS является единственным передатчиком информации, поэтому здесь не возникает конкуренции за доступ к среде. Станция CMTS использует нисходящий канал для передачи по нему кадров данных всем абонентам за счет адресации Ethernet и разделения канала во времени.

Восходящий канал задействуется в режиме множественного доступа всеми кабельными модемами, подключенными к данному абонентскому окончанию. В этой разделяемой среде CMTS играет роль *арбитра*. Каждый абонентский модем начинает передачу только после того, как получит разрешение на это от головного модема по прямому каналу. Для того чтобы один абонентский модем не занимал канал надолго, CMTS назначает каждому абонентскому модему тайм-слот ограниченного размера. Тайм-слоты распределяются только между активными модемами — это позволяет расходовать ограниченную пропускную способность максимально эффективно. Для вновь подключаемых абонентских модемов предназначены специальные тайм-слоты. При включении абонентский модем использует такой тайм-слот, чтобы оповестить CMTS о своем присутствии в сети. Далее он ожидает, когда ему будет выделен тайм-слот на равных основаниях с другими модемами.

Кабельный модем абонента может иметь разъем для подключения обычного телефона, для которого также выделяется полоса в 4 МГц в нижнем диапазоне частот. В этом случае абонент получает от одного поставщика услуг доступ трех типов: телефонный, компьютерный и телевизионный.

Беспроводной доступ

Мы уже касались особенностей беспроводной передачи данных в предыдущих главах: в главе 10 были рассмотрены общие принципы беспроводной связи, а в главе 14 — технологии беспроводных локальных и персональных сетей. Беспроводная передача данных в последнее время широко используется также для организации доступа, особенно в тех случаях, когда поставщик услуг по какой-то причине не может обеспечить своим клиентам проводной доступ. Чаще всего это случается с альтернативными поставщиками услуг, которые не имеют в своем распоряжении проводных абонентских окончаний к домам клиентов. Другим типичным примером является организация временного высокоскоростного доступа для определенного здания, например при проведении конференции в помещении гостиницы, не оснащенном средствами проводного доступа необходимой пропускной способности.

Беспроводной доступ может быть как фиксированным, так и мобильным.

Фиксированный беспроводной доступ организуется для абонентов, компьютеры которых находятся в пределах ограниченной территории, чаще всего в пределах здания. В таком случае поставщик услуг может использовать направленную антенну и передатчик известной мощности, чтобы обеспечить устойчивый прием высокочастотных сигналов в такой узкой области покрытия, как здание. Если у поставщика услуг имеется достаточно большое количество абонентов фиксированного беспроводного доступа, то он обычно задействует

несколько направленных антенн, чтобы покрыть все секторы, в которых находятся его абоненты.

Для беспроводного фиксированного доступа употребляется также термин **беспроводное абонентское окончание** (Wireless Local Loop, WLL). Этот термин хорошо отражает тот факт, что, несмотря на отсутствие кабелей, абоненты «привязаны» к определенной географической точке, как и в случае проводного абонентского окончания.

Существуют *узкополосные* и *широкополосные* беспроводные абонентские окончания. Первый тип не обеспечивает передачу телевизионного сигнала, а только сравнительно низкоскоростной компьютерный трафик (64–128 Кбит/с) и телефонный сигнал. Второй тип обычно основан на системах распространения телевизионного сигнала, поэтому работает с высокочастотными диапазонами и обеспечивает все три вида доступа, причем компьютерные данные передаются обычно со скоростями в несколько сотен килобит в секунду или несколько мегабит в секунду.

К системам последнего типа относятся **многоканальная служба распределения** (Multichannel Multipoint Distribution Service, MMDS) и **локальная служба распределения** (Local Multipoint Distribution Service, LMDS). MMDS работает в диапазоне 2,1 ГГц, а LMDS – 30 ГГц в Америке и 40 ГГц в Европе. Обе системы обеспечивают двунаправленную передачу сигналов для абонентов телевизионных, телефонных и компьютерных услуг. Так как система MMDS работает на существенно более низких частотах, чем LMDS, она обеспечивает гораздо более широкую область покрытия. Одна мачта с направленными антеннами MMDS обычно может обслуживать территорию радиусом в 50 км, в то время как радиус покрытия передатчиков LMDS обычно не превышает 5 км, а в городских условиях он может быть и того меньше. Зато LMDS может обеспечить для своих абонентов более высокие скорости доступа (до 155 Мбит/с).

Как в узкополосных, так и в широкополосных беспроводных абонентских окончаниях используются различные методы мультиплексирования сигналов для одновременной работы своих абонентов в одном секторе направленности антенны, а также для разделения телевизионного, телефонного и компьютерного трафиков. Обычно здесь применяется комбинация приемов FDM и TDM. Например, для каждого типа трафика может быть выделен определенный диапазон частот в соответствии с принципами частотного мультиплексирования. Затем внутри диапазона частот, выделенного для компьютерного трафика, может применяться асинхронное временное мультиплексирование с определенным алгоритмом доступа к общей среде, например с центральным арбитром. Для некоторых абонентов, которым необходима гарантированная полоса пропускания, может применяться синхронное временное мультиплексирование с образованием беспроводных каналов PDH/SDH.

К сожалению, технологии WLL до сих пор во многом являются фирменными с несовместимыми оборудованием доступа и центральными станциями. Для устранения этого недостатка был разработан стандарт **IEEE 802.16** (известный под названием WiMAX), который определяет некоторые общие принципы использования частотного диапазона, методов мультиплексирования и предоставляемые услуги. Также этот стандарт предусматривает применение разнообразных методов мультиплексирования, как частотного, так и временного синхронного и асинхронного, чтобы учесть интересы разных производителей оборудования WLL и обеспечить максимальную гибкость таких систем.

Технология 802.11 также может использоваться для фиксированного беспроводного доступа. Однако она применяется в этом качестве не так часто, потому что ориентирована исключительно на компьютерный трафик и игнорирует особенности телефонного

и телевизионного трафиков, а именно — доступ с постоянной битовой скоростью. Метод доступа CDMA/CA, описываемый в 802.11, не может обеспечить требуемого уровня QoS для чувствительного к задержкам трафика. Тем не менее некоторые поставщики услуг применяют технологию 802.11 для фиксированного доступа в Интернет тех абонентов, которых удовлетворяет неопределенная пропускная способность. Эта технология также популярна для «кочевого» доступа в зонах временного пребывания абонентов, например в аэропортах или на железнодорожных вокзалах.

Беспроводной мобильный доступ в Интернет предоставляется сегодня в основном операторами мобильных телефонных сетей. Мобильная телефония второго поколения обеспечивает доступ в Интернет, используя в качестве транспорта с коммутацией пакетов протокол GPRS (General Packet Radio Service — служба пакетной радиосвязи общего назначения), который работает в сетях D-AMPS и GSM. Однако скорость такого доступа невысока (всего 2400–9800 Кбит/с). В мобильных сетях третьего поколения, которые только начинают разворачиваться, эта скорость должна существенно возрасти (до 2 Мбит/с).

Выводы

Термин «удаленный доступ» применяется в том случае, когда говорят о доступе домашних пользователей или сотрудников мелких филиалов предприятий к ресурсам Интернета или корпоративной сети.

Существуют различные категории клиентов удаленного доступа, отличающиеся используемыми абонентскими окончаниями, наличием или отсутствием домашней локальной сети, требованиями к скорости доступа и типом ресурсов, к которым требуется обеспечить доступ (ресурсы публичного домена Интернета или корпоративной сети).

Поставщик услуг обычно стремится сделать абонентское окончание универсальным, то есть способным передавать трафик трех основных терминальных устройств массового пользователя: телефона, телевизора и компьютера.

Базовым сервисом удаленного доступа является режим удаленного узла, когда компьютер пользователя становится узлом локальной сети поставщика услуг или своего предприятия.

Особым режимом удаленного доступа является удаленное управление, когда компьютер пользователя эмулирует терминал, подключенный к другому компьютеру. Удаленное управление позволяет пользователю получить полный контроль над другим компьютером и запускать на нем любые приложения. Это удобно для пользователя, но представляет большую потенциальную опасность для корпоративных ресурсов.

Наиболее старым видом удаленного доступа является коммутируемый доступ через аналоговые окончания PSTN. С помощью обычного модема компьютер устанавливает в телефонной сети соединение с сервером удаленного доступа, подключенного к сети с коммутацией пакетов.

Фиксированная полоса пропускания в 4 кГц, выделяемая пользователям телефонной сети, принципиально ограничивает скорость передачи обычных модемов. Модемы V.90 обеспечивают восходящую скорость до 33,6 Кбит/с и нисходящую скорость до 56 Кбит/с, но в последнем случае только тогда, когда все транзитные телефонные коммутаторы от клиента до сервера удаленного доступа являются цифровыми.

Технология ISDN была разработана для создания универсальной сети, оказывающей, в том числе, услуги компьютерного доступа. Однако сегодня ее скорость передачи (128 Кбит/с) считается слишком низкой для доступа массовых клиентов к мультимедийной информации.

Технология ADSL полностью использует полосу пропускания телефонного абонентского окончания, деля ее на три канала: дуплексный голосовой, восходящий (до 1 Мбит/с) и нисходящий (до 24 Мбит/с) компьютерные. Ограничение на полосу пропускания для абонента телефонной сети в 4 кГц не влияет

на работу ADSL-модемов, так как компьютерные данные в ближайшей точке присутствия ответвляются в сеть с коммутацией пакетов.

Кабельные модемы работают на коаксиальном абонентском окончании CATV, которое является разделяемой средой для нескольких абонентов, подключенных к одному и тому же кабелю. Широкая полоса пропускания коаксиального кабеля обеспечивает восходящую скорость до 10 Мбит/с, а нисходящую — до 30–40 Мбит/с.

Для фиксированного беспроводного доступа служит множество фирменных технологий, обеспечивающих доставку пользователю телефонной, телевизионной и компьютерной информации. Для предоставления разнообразных услуг такой доступ требует сочетания частотного и временно́го мультиплексирования, а также коммутации каналов и пакетов.

Мобильный доступ пока существует в виде дополнительной низкоскоростной услуги по передаче данных через сотовые телефонные сети второго поколения. Стандарты сетей третьего поколения предусматривают более высокие скорости передачи данных, но их внедрение только начинается.

Вопросы и задания

1. Каким образом может оказывать услуги доступа поставщик услуг, который не владеет кабельными абонентскими окончаниями? Варианты ответов:
   а) арендовать абонентские окончания у провайдера, которому они принадлежат;
   б) заключить с провайдером, которому принадлежат абонентские окончания, договор на направление трафика пользователей в свою сеть;
   в) организовать радиодоступ для клиентов.
2. Какие характеристики клиентов удаленного доступа нужно принимать во внимание при организации такого доступа?
3. Какое абонентское окончание можно назвать универсальным?
4. В режиме удаленного управления:
   а) компьютер пользователя работает как монитор удаленного компьютера;
   б) невозможно обмениваться файлами с удаленным компьютером;
   в) локальный и удаленный компьютеры являются равноправными.
5. Какой вид доступа используется при конфигурировании маршрутизаторов?
6. Почему скорости обычных (коммутируемых) модемов намного уступают скоростям ADSL-модемов и кабельных модемов? Варианты ответов:
   а) коммутируемый модем в отличие от ADSL-модема не поддерживает параллельную обработку кадров;
   б) полоса пропускания, доступная коммутируемому модему, существенно уже;
   в) коммутируемый модем использует телефонную сеть «из конца в конец», а ADSL-модем — нет.
7. К устройству какого уровня в терминах модели OSI можно отнести модем?
8. Чем отличаются требования к локальной сети провайдера, предоставляющего услуги коммутируемого доступа, от требований к локальной сети поставщика услуг доступа через ADSL? Варианты ответов:
   а) локальная сеть поставщика услуг ADSL должна иметь существенно более высокую пропускную способность;

б) локальная сеть поставщика услуг ADSL должна быть разбита на несколько сетей VLAN;

в) ничем.

9. Какой метод доступа к разделяемой среде используют кабельные модемы? Варианты ответов:

   а) CSMA/CD;

   б) CSMA/CA;

   в) централизованный доступ, управляемый арбитром.

10. Вы купили модем V.90 и связываетесь по телефонной сети со своим знакомым, который также использует модем V.90. Вы уверены, что все телефонные коммутаторы на пути между вами и вашим знакомым работают в цифровом режиме. На какой скорости вы получите соединение со своим знакомым?

11. Какую услугу ISDN целесообразно использовать, если к сети ISDN подключены с помощью маршрутизаторов две локальные сети, причем межсетевой трафик в течение длительного периода времени имеет интенсивность от 100 до 512 Кбит/с? Варианты ответов:

    а) постоянное соединение по интерфейсу PRI;

    б) коммутируемое соединение по интерфейсу BRI;

    в) постоянное соединение по двум интерфейсам BRI.

12. Что необходимо изменить в настройке, если ADSL-модем работает на абонентском окончании с недопустимо высоким процентом ошибок? Варианты ответов:

    а) снизить скорость передачи данных;

    б) повысить скорость передачи данных;

    в) увеличить полосу пропускания линии.

ГЛАВА 23 Сетевые службы

С точки зрения пользователей компьютерные сети представляют собой набор служб (сервисов), таких как электронная почта, WWW, интернет-телефония и интернет-телевидение. Транспортные функции сети, обеспечивающие работу этих служб, скрыты от пользователей, хотя иногда и влияют на некоторые детали предоставления службы, например, недостаточно высокая надежность доступа в Интернет по телефонным каналам потребовала коротких TCP-сеансов в службе WWW при передаче содержания веб-страниц. Помимо служб, ориентированных на конечных пользователей, существуют службы, ориентированные на сетевых администраторов, решающих задачи конфигурирования и управления сетевыми устройствами; в эту категорию входят службы FTP, telnet[1] и SNMP. Дополняют общую картину уже рассмотренные нами службы, помогающие компьютерам и сетевым устройствам организовать свою работу, такие как службы DNS и DHCP.

[1] Краткий обзор функций протокола telnet был приведен в главе 22.

Электронная почта

Сетевая почтовая служба, или **электронная почта**, — это распределенное приложение, главной функцией которого является предоставление пользователям сети возможности обмениваться электронными сообщениями.

Как и все сетевые службы, электронная почта построена в архитектуре клиент-сервер. Почтовый клиент всегда располагается на компьютере пользователя, а почтовый сервер, как правило, работает на выделенном компьютере.

Почтовый клиент (называемый также **агентом пользователя**) — это программа, предназначенная для поддержания пользовательского интерфейса (обычно графического), а также для предоставления пользователю широкого набора услуг по подготовке электронных сообщений. В число таких услуг входит создание текста в различных форматах и кодировках, сохранение, уничтожение, переадресация, сортировка писем по разным критериям, просмотр перечня поступивших и отправленных писем, грамматическая и синтаксическая проверка текста сообщений, ведение адресных баз данных, автоответы, образование групп рассылки и прочее, и прочее. Кроме того, почтовый клиент поддерживает взаимодействие с серверной частью почтовой службы.

Почтовый сервер выполняет прием сообщений от клиентов, для чего он постоянно находится в активном состоянии. Кроме того, он выполняет буферизацию сообщений, распределение поступивших сообщений по индивидуальным буферам (почтовым ящикам) клиентов, управляет объемами памяти, выделяемой клиентам, выполняет регистрацию клиентов и регламентирует их права доступа к сообщениям, а также решает много других задач.

Электронные сообщения

Почтовая служба оперирует **электронными сообщениями** — информационными структурами определенного стандартного формата. Упрощенно электронное сообщение может быть представлено в виде двух частей, одна из которых (заголовок) содержит вспомогательную информацию для почтовой службы, другая часть (тело сообщения) — это собственно то «письмо», которое предназначается для прочтения, прослушивания или просмотра адресатом (RFC 8022).

Главными элементами заголовка являются адреса отправителя и получателя в виде Polina@domen.com, где Polina — идентификатор пользователя почтовой службы, а domen.com — имя домена, к которому относится этот пользователь. Кроме этого, почтовая служба включает в заголовок дату и тему письма, делает отметки о применении шифрования, срочности доставки, необходимости подтверждения факта прочтения этого сообщения адресатом и др. Дополнительная информация заголовка может оповещать почтового клиента получателя об использовании той или иной кодировки. Помимо основной кодировки ASCII, современные почтовые системы позволяют создавать сообщения, включающие изображения (в форматах GIF и JPEG), а также аудио- и видеофайлы.

Протокол SMTP

В качестве средств передачи сообщения почтовая служба использует стандартный, разработанный специально для почтовых систем протокол **SMTP** (Simple Mail Transfer Protocol —

простой протокол передачи почты). Как и большинство других протоколов прикладного уровня, SMTP реализуется несимметричными взаимодействующими частями: SMTP-клиентом и SMTP-сервером. Важно отметить, что этот протокол *ориентирован на передачу данных по направлению от клиента к серверу*, следовательно, SMTP-клиент работает на стороне отправителя, а SMTP-сервер — на стороне получателя. SMTP-сервер должен постоянно быть в режиме подключения, ожидая запросов со стороны SMTP-клиента.

Логика работы протокола SMTP действительно является достаточно простой (как это и следует из его названия). После того как, применяя графический интерфейс своего почтового клиента, пользователь щелкает на значке, инициирующем отправку сообщения, SMTP-клиент посылает запрос на установление TCP-соединения на порт 25 (это назначенный порт SMTP-сервера). Если сервер готов, то он посылает свои идентифицирующие данные, в частности свое DNS-имя. Затем клиент передает серверу адреса (имена) отправителя и получателя. Если имя получателя соответствует ожидаемому, то после получения адресов сервер дает согласие на установление TCP-соединения, и в рамках этого надежного логического канала происходит передача сообщения. Используя одно TCP-соединение, клиент может передать несколько сообщений, предваряя каждое из них указанием адресов отправителя и получателя. После завершения передачи TCP- и SMTP-соединения разрываются. Если в начале сеанса связи SMTP-сервер оказался не готов, то он посылает соответствующее сообщение клиенту, в ответ тот снова посылает запрос, пытаясь заново установить соединение. Если сервер не может доставить сообщение, то он передает отчет об ошибке отправителю сообщения и разрывает соединение. После того как передача сообщения благополучно заканчивается, переданное сообщение сохраняется в буфере на сервере.

ПРИМЕЧАНИЕ

Хотя в любом протоколе предполагается обмен данными между взаимодействующими частями, то есть данные передаются в обе стороны, различают протоколы, ориентированные на передачу (pull protocols), и протоколы, ориентированные на прием данных (push protocols). В протоколах, ориентированных на передачу, к которым, в частности, относится протокол SMTP, клиент является инициатором передачи данных на сервер, а в протоколах, ориентированных на прием, к которым относятся, например, протоколы HTTP, POP3 и IMAP, клиент является инициатором получения данных от сервера.

Непосредственное взаимодействие клиента и сервера

Теперь, когда мы обсудили основные составляющие почтовой службы, давайте рассмотрим несколько основных схем ее организации. Начнем с простейшего, практически не используемого сейчас варианта, когда отправитель непосредственно взаимодействует с получателем. Как показано на рис. 23.1, у каждого пользователя на компьютере устанавливаются почтовые клиент и сервер.

Данила, используя графический интерфейс своего почтового клиента, вызывает функцию создания сообщения, в результате на экране появляется стандартная незаполненная форма сообщения, в поля которой Данила вписывает свой адрес, адрес Полины и тему письма, а затем набирает текст письма. При этом он может пользоваться не только встроенным в почтовую программу текстовым редактором, но и привлекать для этой

цели другие программы, например MS Word. Когда письмо готово, Данила вызывает функцию отправки сообщения, и встроенный SMTP-клиент посылает запрос на установление связи SMTP-серверу на компьютере Полины. В результате устанавливаются SMTP- и TCP-соединения, и сообщение передается через сеть. Почтовый сервер Полины сохраняет письмо в памяти ее компьютера, а почтовый клиент по команде Полины выводит его на экран, при необходимости выполняя преобразование формата. Полина может сохранить, переадресовать или удалить это письмо. Понятно, что в том случае, когда Полина решит направить электронное сообщение Даниле, схема работы почтовой службы будет симметричной.

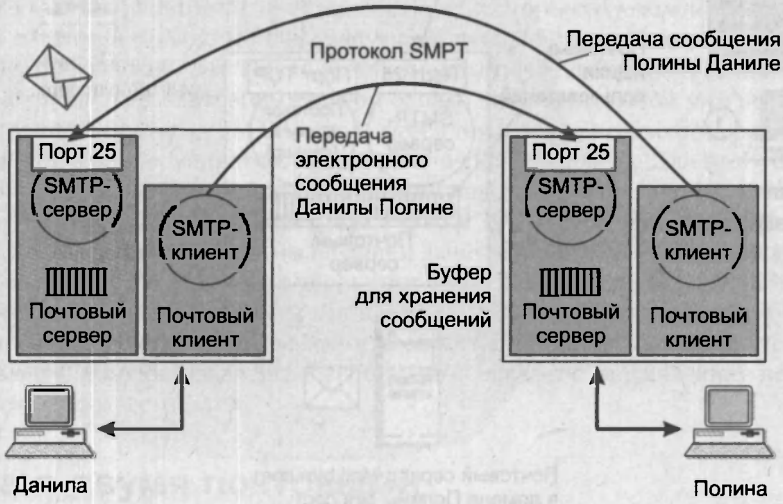


Рис. 23.1. Схема непосредственного взаимодействия клиента и сервера

Схема с выделенным почтовым сервером

Рассмотренная только что простейшая схема почтовой связи кажется работоспособной, однако у нее есть серьезный и очевидный дефект. Мы упоминали, что для обмена сообщениями необходимо, чтобы SMTP-сервер постоянно находился в ожидании запроса от SMTP-клиента. Это означает, что для того чтобы письма, направленные Полине, доходили до нее, ее компьютер должен постоянно находиться в режиме подключения. Понятно, что такое требование для многих пользователей неприемлемо.

Естественным решением этой проблемы является размещение SMTP-сервера на специально выделенном для этой цели *компьютере-посреднике*. Это должен быть достаточно мощный и надежный компьютер, способный круглосуточно передавать почтовые сообщения от многих отправителей ко многим получателям. Обычно почтовые серверы поддерживаются крупными организациями для своих сотрудников или провайдерами для своих клиентов.

Для каждого домена имен система DNS создает записи типа MX, в которых хранятся DNS-имена почтовых серверов, обслуживающих пользователей, относящихся к этому домену.

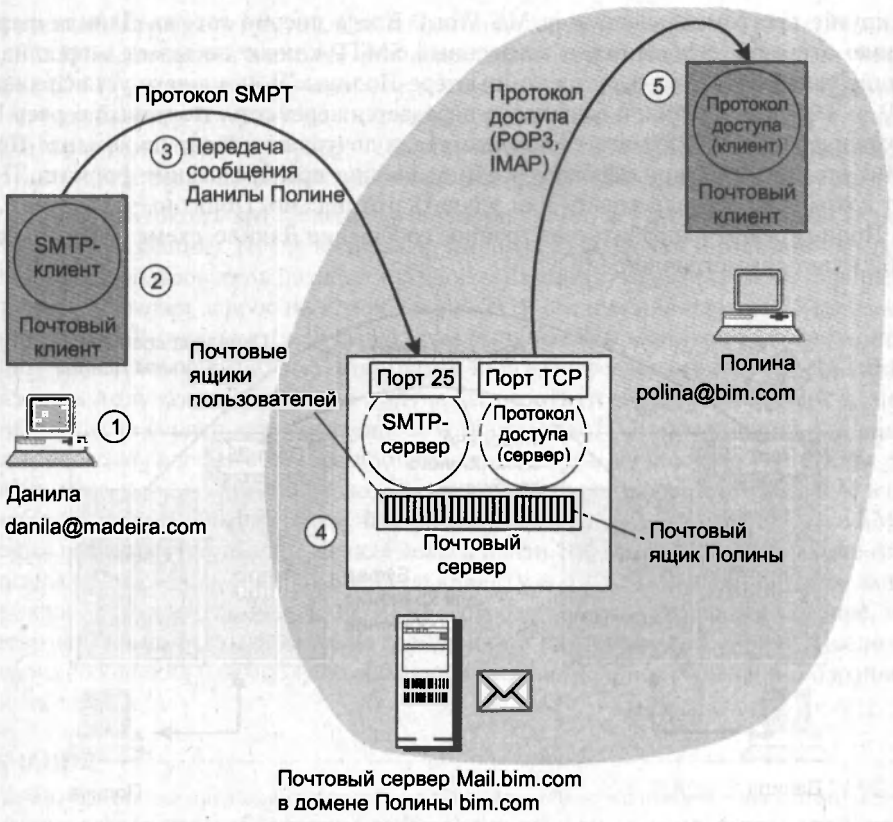


Рис. 23.2. Схема с выделенным почтовым сервером в принимающем домене

На рис. 23.2 представлена схема с выделенным почтовым сервером. Чтобы не усложнять рисунок, мы показали на нем только те компоненты, которые участвуют в передаче сообщения от Данилы к Полине. Для обратного случая схема должна быть симметрично дополнена.

1. Итак, пусть Данила решает послать письмо Полине, для чего он запускает на своем компьютере установленную на нем программу почтового клиента (например, Microsoft Outlook или Mozilla Thunderbird). Он пишет текст сообщения, указывает необходимую сопроводительную информацию, в частности адрес получателя polina@bim.com, и щелкает мышью на значке отправки сообщения. Поскольку готовое сообщение должно быть направлено совершенно определенному почтовому серверу, клиент обращается к системе DNS, чтобы определить имя почтового сервера, обслуживающему домен Полины bim.com. Получив от DNS в качестве ответа имя mail.bim.com, SMTP-клиент еще раз обращается к DNS, на этот раз, чтобы узнать IP-адрес почтового сервера mail.bim.com.
2. SMTP-клиент посылает по данному IP-адресу запрос на установление TCP-соединения через порт 25 (SMTP-сервер).
3. С этого момента начинается диалог между клиентом и сервером по протоколу SMTP, с которым мы уже знакомы. Заметим, что здесь, как и у всех протоколов, ориентированных на передачу, направление передачи запроса от клиента на установление SMTP-

соединения совпадает с направлением передачи сообщения. Если сервер оказывается готовым, то после установления TCP-соединения сообщение Данилы передается.

4. Письмо сохраняется в буфере почтового сервера, а затем направляется в индивидуальный буфер, отведенный системой для хранения корреспонденции Полины. Такого рода буферы называют почтовыми ящиками. Важно заметить, что помимо Полины у почтового сервера имеется еще много других клиентов, и это усложняет его работу. То есть почтовый сервер должен решать самые разнообразные задачи по организации многопользовательского доступа, включая управление разделяемыми ресурсами и обеспечение безопасного доступа.
5. В какой-то момент, который принципиально *не связан с моментом поступления сообщений* на почтовый сервер, Полина, запускает свою почтовую программу и выполняет команду проверки почты. После этой команды почтовый клиент должен запустить протокол доступа к почтовому серверу, однако этим протоколом уже не будет SMTP. Напомним, что протокол SMTP используется тогда, когда необходимо передать данные на сервер, а Полине, напротив, нужно получить их с сервера. Для этого случая были разработаны другие протоколы, обобщенно называемые протоколами доступа к почтовому серверу, такие, например, как POP3 и IMAP. Оба этих протокола относятся к протоколам, ориентированным на прием данных (протокол POP3 ожидает запрос на установление TCP-соединения через порт 110, а IMAP — через порт 143, на рисунке эти порты обобщенно показаны как порт TCP). В результате работы любого из них письмо Данилы оказывается в памяти компьютера Полины. Заметим, что на этот раз направление запроса от клиента к серверу не совпадает с направлением передачи данных, показанному стрелкой.

Схема с двумя почтовыми серверами-посредниками

Прежде чем мы перейдем к сравнению двух протоколов доступа к почте, давайте посмотрим на еще одну схему организации почтовой службы, наиболее приближенную к реальности (рис. 23.3). Здесь передача сообщений между клиентами почты (на нашем рисунке между отправителем Данилой и получателем Полиной) проходит через два промежуточных почтовых сервера, каждый из которых обслуживает домен своего клиента. На каждом из этих серверов установлены также и клиентские части протокола SMTP. При отправке письма почтовый клиент Данилы передает сообщение по протоколу SMTP почтовому серверу домена, к которому относится Данила — RoyalMail.madeira.com. Это сообщение буферизуется на данном сервере, а затем по протоколу SMTP передается дальше на почтовый сервер домена Полины — mail.bim.com, откуда описанным уже образом попадает на компьютер Полины.

Возникает вопрос, зачем нужна такая двухступенчатая передача через два почтовых сервера? Прежде всего, для повышения надежности и гибкости процедуры доставки сообщения. Действительно, в схеме с передачей сообщения сразу на сервер получателя почтовый клиент отправителя в случае неисправности почтового сервера должен самостоятельно справляться со сложившейся нештатной ситуацией. Если же посредником в передаче сообщения является другой почтовый сервер, то это позволяет реализовывать разнообразные логические механизмы реакции на отказы на стороне сервера, который к тому же всегда находится в режиме подключения. Например, при невозможности передать письмо почто-

вому серверу получателя сервер отправляющей стороны может не только рапортовать об этом своему клиенту, но и предпринимать собственные действия — пытаться снова и снова послать письмо, повторяя эти попытки в течение достаточно длительного периода.

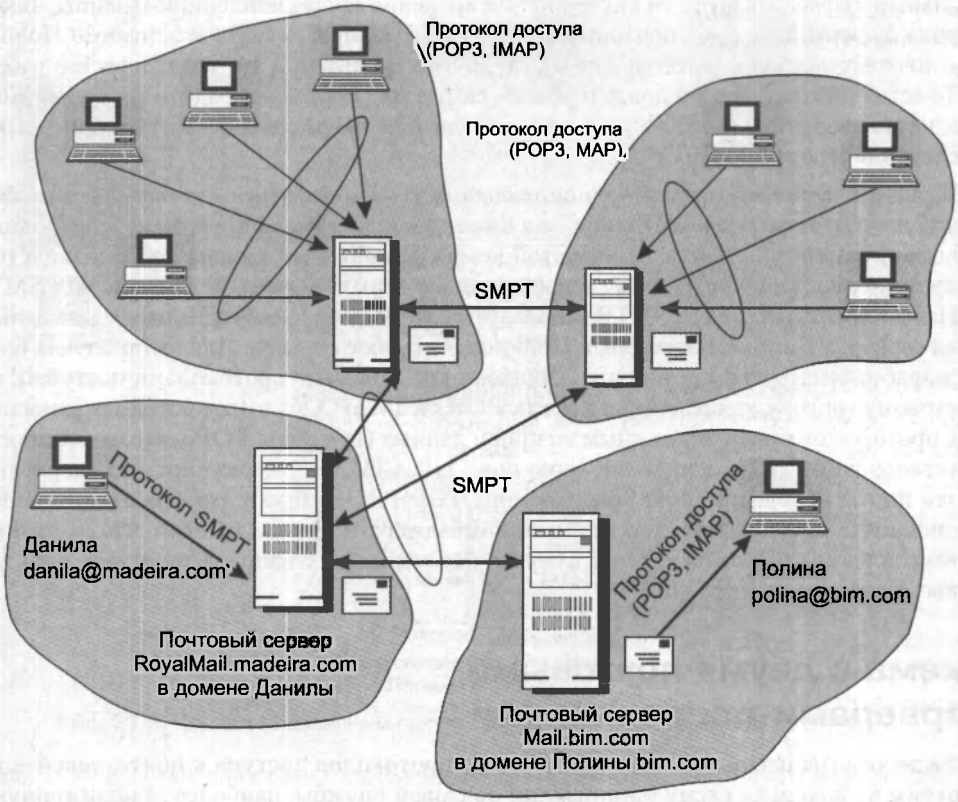


Рис. 23.3. Схема с выделенными почтовыми серверами в каждом домене

Протоколы POP3 и IMAP

А теперь давайте, как мы и собирались, сравним два протокола доступа к почте: POP3 (Post Office Protocol v.3 — протокол почтового отделения версии 3) и IMAP (Internet Mail Access Protocol — протокол доступа к электронной почте Интернета). Оба эти протокола решают одну и ту же задачу — обеспечивают доступ пользователей к корреспонденции, хранящейся на почтовом сервере. В связи с многопользовательским характером работы почтового сервера оба протокола поддерживают аутентификацию пользователей на основе идентификаторов и паролей. Однако протоколы POP3 и IMAP имеют принципиальные различия, важнейшее из которых состоит в следующем. Получая доступ к почтовому серверу по протоколу POP3, вы «перекачиваете» адресованные вам сообщения в память своего компьютера, при этом на сервере не остается никакого следа от считанной вами почты. Если же доступ осуществляется по протоколу IMAP, то в память вашего компьютера передаются только копии сообщений, хранящихся на почтовом сервере.

Это различие серьезно влияет на характер работы с электронной почтой. Сейчас очень распространенной является ситуация, когда человек в течение одного и того же периода времени использует несколько различных компьютеров: на постоянном месте работы, дома, в командировке. Теперь давайте представим, что произойдет с корреспонденцией пользователя Полины, если она получает доступ к почте по протоколу POP3. Письма, прочитанные на работе, останутся в памяти ее рабочего компьютера. Придя домой, она уже не сможет прочитать их снова. Опросив почту дома, она получит все сообщения, которые поступили с момента последнего обращения к почтовому серверу, но из памяти сервера они исчезнут, и завтра на работе она, возможно, не обнаружит важные служебные сообщения, которые были загружены на диск ее домашнего ноутбука. Таким образом, получаемая Полиной корреспонденция будет «рассеяна» по всем компьютерам, которыми она пользовалась. Такой подход не позволяет рационально организовать почту: распределять письма по папкам, сортировать их по разным критериям, отслеживать состояние переписки, отмечать письма, на которые послан ответ, и письма, еще требующие ответа и т. д. Конечно, если пользователь всегда работает только с одним компьютером, недостатки протокола POP3 не являются столь критичными. Но и в этом случае проявляется еще один «дефект» этого протокола — клиент не может пропустить, не читая, ни одного письма, поступающего от сервера. То есть объемное и возможно совсем ненужное вам сообщение может надолго заблокировать вашу почту.

Протокол IMAP был разработан как ответ на эти проблемы. Предположим, что теперь Полина получает почту по протоколу IMAP. С какого компьютера она бы ни обратилась к почтовому серверу, ей будут переданы только копии запрошенных сообщений. Вся совокупность полученной корреспонденции останется в полной сохранности в памяти почтового сервера (если, конечно, не поступит специальной команды от пользователя об удалении того или иного письма). Такая схема доступа делает возможным для сервера предоставление широкого перечня услуг по рациональному ведению корреспонденции, то есть именно того, чего лишен пользователь при применении протокола POP3. Важным преимуществом IMAP является также возможность предварительного чтения заголовка письма, после чего пользователь может принять решение о том, есть ли смысл получать с почтового сервера само письмо.

Веб-служба

Изобретение службы World Wide Web (WWW), или Всемирной паутины, стоит в одном ряду с изобретениями телефона, радио и телевидения. Благодаря WWW люди получили возможность доступа к нужной им информации в любое удобное для них время. Теперь проще найти интересующую вас статью в Интернете, чем в стопке журналов, хранящихся рядом в шкафу. Очень быстро исчезают многие традиционные приемы рациональной организации работы с информацией, заключающиеся, например, в хранении полезной информации в записных книжках, раскладывании вырезок из журналов и газет в картонные папки с веревочками, упорядочивании документов в каталогах путем наклеивания на них маркеров с условными кодами, помогающими быстро отыскать нужный документ и т. д. Этим приемам приходят на смену новые безбумажные технологии Интернета, среди которых важнейшей является сетевая служба WWW, или веб-служба. Заметим, что WWW не только предоставляет любому человеку возможность быстрого поиска нужных данных и доступа к ним, но и позволяет выносить на многомиллионную аудиторию пользователей

Интернета собственную информацию — мнения, художественные и публицистические произведения, результаты научной работы, объявления и т. д. Причем он может это делать без особых организационных забот и практически бесплатно.

Мы не будем долго останавливаться на описании всех возможностей этой службы, учитывая, что для большинства из нас регулярный просмотр веб-сайтов стал не просто обыденностью, а необходимым элементом жизненного уклада.

Веб- и HTML-страницы

Миллионы компьютеров, связанных через Интернет, хранят невообразимо огромные объемы информации, представленной в виде веб-страниц.

- **Веб-страница**, или **веб-документ**, как правило, состоит из основного HTML-файла и некоторого количества ссылок на другие объекты разного типа: JPEG- и GIF-изображения, другие HTML-файлы, аудио- и видеофайлы.
- **HTML-файлом**, **HTML-страницей** или **гипертекстовой страницей** называют файл, который содержит текст, написанный на языке HTML (HyperText Markup Language — язык разметки гипертекста).

История появления языка HTML связана с попытками программистов разработать средство, которое бы позволяло им программным путем создавать красиво сверстанные страницы для просмотра на экране. Другими словами, красивая картинка появляется на дисплее только в результате ее интерпретации специальной программой, а в исходном виде она представляет собой однообразный текст с множеством служебных пометок. Вместо применения различных приемов форматирования, таких как выделение заголовков крупным шрифтом, а важных выводов — курсивным или полужирным начертанием, создатель документа на языках этого типа просто вставляет в текст соответствующие указания о том, что данная часть текста должна быть выведена на экран в том или ином виде. Служебные пометки такого рода в исходном тексте выглядят, например, как `<b> </b>` (начать и закончить вывод текста полужирным начертанием) и называются **тегами**. Язык HTML не является первым языком разметки текста, его предшественники существовали задолго до появления веб-службы, например в первых версиях ОС Unix существовал язык troff (с помощью этого языка отформатированы страницы электронной документации Unix, известные как шап-страницы).

В язык HTML включены разные типы тегов, команд и параметров, в том числе для вставки в текст изображений (тег `<img src="...">`). Чтобы HTML-страница выглядела так, как задумал программист, она должна быть выведена на экран специальной программой, способной интерпретировать язык HTML. Такой программой является уже упоминавшийся веб-браузер.

Существует особый тип тега, который имеет вид `<a href="..."> ...</a>` и называется **гиперссылкой**. Гиперссылка содержит информацию о веб-странице или объекте, который может находиться как на том же компьютере, так и на других компьютерах Интернета. Отличие гиперссылки от других тегов состоит в том, что элемент, описываемый ею, не появляется автоматически на экране, вместо этого на месте тега (гиперссылки) на экран выводится некоторое условное изображение или особым образом выделенный текст — имя гиперссылки. Чтобы получить доступ к объекту, на который указывает эта гиперссылка, пользователь

должен «щелкнуть» на ней, дав тем самым команду браузеру найти и вывести на экран требуемую страницу или объект. После того как новая веб-страница будет загружена, пользователь сможет перейти по следующей гиперссылке — такой «веб-серфинг» может продолжаться теоретически сколь угодно долго. Все это время веб-браузер будет находить указанные в гиперссылках страницы, интерпретировать все размещенные на них указания и выводить информацию на экран в том виде, в котором ее спроектировали разработчики этих страниц.

URL

Браузер находит веб-страницы и отдельные объекты по адресам специального формата, называемым **URL** (Uniform Resource Locator — унифицированный указатель ресурса). URL-адрес может выглядеть, например, так: http://www.olifer.co.uk/books/books.htm.

В URL-адресе можно выделить три части:

- *Тип протокола доступа.* Начальная часть URL (http://) указывает на то, какой протокол должен быть использован для доступа к данным, расположение которых определяется оставшейся частью URL. Помимо HTTP, здесь могут быть указаны и другие протоколы, такие как FTP, telnet, также позволяющие осуществлять удаленный доступ к файлам или компьютерам[1].
- *DNS-имя сервера.* Имя сервера, на котором хранится нужная страница. В нашем случае — это имя сайта www.olifer.co.uk.
- *Путь к объекту.* Обычно это составное имя файла (объекта) относительно главного каталога веб-сервера, предлагаемого по умолчанию. В нашем случае путем к объекту является /books/books.htm. По расширению файла мы можем сделать вывод о том, что это HTML-файл.

Веб-клиент и веб-сервер

Как мы уже отмечали, сетевая веб-служба представляет собой распределенную программу, построенную в архитектуре клиент-сервер. Клиент и сервер веб-службы взаимодействуют друг с другом по протоколу HTTP.

Клиентская часть веб-службы, или **веб-клиент**, называемый также **браузером**, или **агентом пользователя** веб-службы, представляет собой приложение, которое устанавливается на компьютере конечного пользователя и одной из важных функций которого является поддержание графического пользовательского интерфейса.

Через этот интерфейс пользователь получает доступ к широкому набору услуг, главной из которых, конечно, является «веб-серфинг», включающий поиск и просмотр страниц, навигацию между уже просмотренными страницами, переход по закладкам и хранение истории посещений. Помимо средств просмотра и навигации, веб-браузер предоставляет пользователю возможность манипулирования страницами: сохранение их в файле на диске своего компьютера, вывод на печать, передача по электронной почте, контекстный поиск

[1] Если в URL-адресе не указывается тип протокола доступа, то браузер по умолчанию использует протокол HTTP.

в пределах страницы, изменение кодировки и формата текста, а также множество других функций, связанных с представлением информации на экране и конфигурированием самого браузера.

К числу наиболее популярных сейчас браузеров можно отнести Internet Explorer компании Microsoft, Firefox компании Mozilla и последнее предложение компании Google — Chrome. Веб-браузер — это не единственный вид клиента, который может обращаться к веб-серверу. Эту роль могут исполнять любые программы и устройства, поддерживающие протокол HTTP, а также многие модели мобильных телефонов — для доступа в этом случае применяется специальный протокол WAP (Wireless Application Protocol — протокол беспроводных приложений).

Значительную часть своих функций браузер выполняет в тесной кооперации с веб-сервером. Как уже было сказано, клиент и сервер веб-службы связываются через сеть по протоколу HTTP. Это означает, что в клиентской части веб-службы присутствует клиентская часть HTTP, а в серверной — серверная часть HTTP.

Веб-сервер — это программа, хранящая объекты локально в каталогах компьютера, на котором она запущена, и обеспечивающая доступ к этим объектам по URL-адресам. Наиболее популярными веб-серверами сейчас являются Apache и Microsoft Internet Information Server.

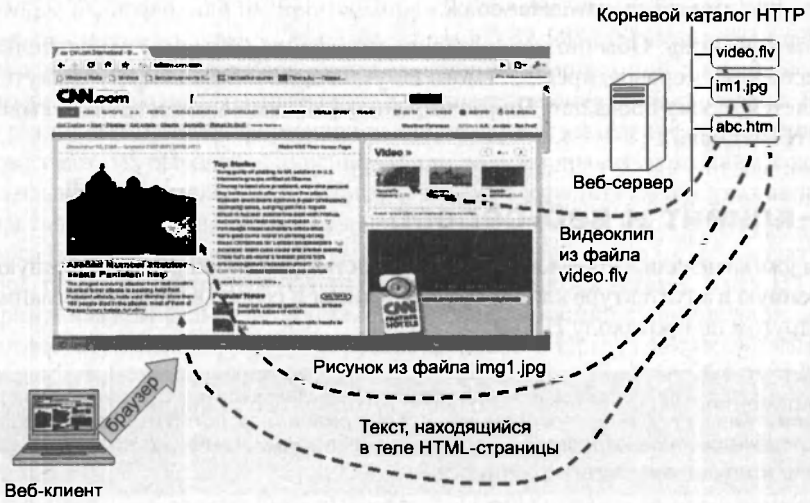


Рис. 23.4. Отображение веб-страницы

Как и любой другой сервер, веб-сервер должен быть постоянно в активном состоянии, прослушивая *TCP-порт 80*, который является назначенным портом протокола HTTP. Как только сервер получает запрос от клиента, он устанавливает TCP-соединение и получает

от клиента имя объекта, например, в виде /books/books.htm, после чего находит в своем каталоге этот файл, а также другие связанные с ним объекты и отсылает по TCP-соединению клиенту. Получив объекты от сервера, веб-браузер отображает их на экране (рис. 23.4). После отправки всех объектов страницы клиенту сервер разрывает с ним TCP-соединение. В дополнительные функции сервера входят также аутентификация клиента и проверка прав доступа данного клиента к данной странице.

Для повышения производительности некоторые веб-серверы прибегают к кэшированию наиболее часто используемых в последнее время страниц в своей памяти. Когда приходит запрос на какую-либо страницу, сервер, прежде чем считывать ее с диска, проверяет, не находится ли она в буферах более «быстрой» оперативной памяти. Кэширование страниц осуществляется и на стороне клиента, а также на промежуточных серверах (прокси-серверах). Кроме того, эффективность обмена данными с клиентом иногда повышают путем компрессии (сжатия) передаваемых страниц. Объем передаваемой информации уменьшают также за счет того, что клиенту передается не весь документ, а только та часть, которая была изменена. Все эти приемы повышения производительности веб-службы реализуются средствами протокола HTTP

Протокол HTTP

HTTP (HyperText Transfer Protocol — протокол передачи гипертекста)[1] — это протокол прикладного уровня, во многом аналогичный протоколам FTP и SMTP. В настоящее время используются две версии протокола HTTP 1.0 и HTTP 1.1

Обмен сообщениями идет по обычной схеме «запрос-ответ». Клиент и сервер обмениваются *текстовыми* сообщениями стандартного формата, то есть каждое сообщение представляет собой нескольких строк обычного текста в кодировке ASCII.

Для транспортировки HTTP-сообщений служит протокол TCP. При этом TCP-соединения могут использоваться двумя разными способами:

- *Долговременное соединение* — передача в одном TCP-соединении нескольких объектов, причем время существования соединения определяется при конфигурировании веб-службы.
- *Кратковременное соединение* — передача в рамках одного TCP-соединения только одного объекта.

Долговременное соединение, в свою очередь, может быть использовано двумя способами:

- *Последовательная передача запросов с простоями* — новый запрос посылается только после получения ответа.
- *Конвейерная передача* — это более эффективный способ, в котором следующий запрос посылается до прибытия ответа на один или несколько предыдущих запросов (напоминает метод скользящего окна). Обычно по умолчанию степень параллелизма устанавливается на уровне 5–10, но у пользователя имеется возможность изменять этот параметр при конфигурировании клиента.

В HTTP 1.1 по умолчанию применяются постоянные соединения и конвейерный режим.

---

[1] RFC 1945, 2616.

Формат HTTP-сообщений

В протоколе HTTP все сообщения состоят из текстовых строк. Сообщения как запросов, так и ответов имеют единую обобщенную структуру из трех частей: обязательной стартовой строки, а также необязательных заголовков и тела сообщения. В табл. 5.1 приведены форматы и примеры стартовых строк и заголовков для запросов и ответов.

Таблица 5.1. Форматы стартовых строк и заголовков

Обобщенная структура сообщения	HTTP-запрос	HTTP-ответ
Стартовая строка (всегда должна быть первой строкой сообщения; обязательный элемент)	Формат запроса Метод/ URL HTTP/1.x. Пример: GET /books/ books.htm HTTP/1.1	Формат ответа: HTTP/1.x КодСостояния Фраза. Пример: HTTP/1.0 200 OK
Заголовки (следуют в произвольном порядке; могут отсутствовать)	Заголовок о DNS-имени компьютера, на котором расположен веб-сервер. Пример: Host: www.olifer.co.uk	Заголовок о времени отправления данного ответа. Пример: Date: 1 Jan 2009 14:00:30
	Заголовок об используемом браузере. Пример: User-agent: Mozilla/5.0	Заголовок об используемом вебсервере. Пример: Server: Apache/1.3.0 (Unix)
	Заголовок о предпочтительном языке. Пример: Accept-language: ru	Заголовок о количестве байтов в теле сообщения. Пример: Content-Length: 1234
	Заголовок о режиме соединения. Пример: Connection: close	Заголовок о режиме соединения. Пример: Connection: close
Пустая строка		
Тело сообщения (может отсутствовать)	Здесь могут быть расположены ключевые слова для поисковой машины или страницы для передачи на сервер	Здесь может быть расположен текст запрашиваемой страницы

Как видно из таблицы, запросы и ответы имеют разные форматы стартовой строки. Каждая из них состоит из трех элементов, включающих поле *версии протокола HTTP*. И в запросе, и в ответе примера указана версия HTTP 1.1. Стартовая строка запроса включает в себя поле *метода* — это название операции, которая должна быть выполнена. Чаще всего в запросах используется метод GET, то есть запрос объекта. Именно он включен в наш пример запроса. Помимо этого метода в запросах протокол предусматривает и другие методы, такие как POST, который используется клиентом, например, для отправки электронной почты или в поисковых машинах, когда клиент запрашивает у сервера не определенный объект, а объекты, содержащие ключевые слова, помещенные в теле сообщения. Еще одним элементом стартовой строки является *URL-ссылка* на запрашиваемый объект — здесь это имя файла /books/books.htm.

В стартовой строке ответа, помимо уже упоминавшегося указания на версию протокола HTTP, имеется поле *кода состояния* и поле *фразы* для короткого текстового сообщения, поясняющего данный код пользователю.

В настоящее время стандарты определяют пять классов кодов состояния:

- 1*xx* — информация о процессе передачи;
- 2*xx* — информация об успешном принятии и обработки запроса клиента (в таблице в примере стартовой строки ответа приведен код и соответствующая фраза 200 OK сообщает клиенту, что его запрос успешно обработан);

- 3*xx* — информация о том, что для успешного выполнения операции нужно произвести следующий запрос по другому URL-адресу, указанному в дополнительном заголовке `Location`;
- 4*xx* — информация об ошибках на стороне клиента (читатель наверняка не раз сталкивался с ситуацией, когда при указании адреса несуществующей страницы браузер выводил на экран сообщение `404 Not Found`);
- 5*xx* — информация о неуспешном выполнения операции по вине сервера (например, сообщение `505 http Version Not Supported` говорит о том, что сервер не поддерживает версию HTTP, предложенную клиентом).

Среди кодов состояния имеется код 401, сопровождаемый сообщением `authorization required`. Если клиент получает такое сообщение в ответ на попытку доступа к странице или объекту, это означает, что доступ к данному ресурсу ограничен и требует авторизации[1] пользователя. Помимо поясняющей фразы сервер помещает в свой ответ дополнительный заголовок `www-Authenticate:<...>`, который сообщает клиенту, какую информацию он должен направить серверу для того, чтобы процедура авторизации могла быть выполнена. Обычно это имя и пароль. Веб-клиент с момента получения такого ответа сервера начинает добавлять во все свои запросы к ресурсам данного сервера дополнительный заголовок `Authorization: <`*имя*, *пароль*`>`, который содержит информацию, необходимую для авторизации доступа.

Динамические веб-страницы

До сих пор мы подразумевали, что содержание страницы не изменяется в зависимости от действий пользователя. Когда пользователь щелкает на гиперссылке, то он переходит на *новую* страницу, а если выполняет команду возвращения обратно, то на экране снова появляется предыдущая страница в *неизменном* виде. Такие страницы называются **статическими**.

Однако в некоторых случаях было бы очень желательно, чтобы содержание страницы изменялось в зависимости от действий пользователя, например при наведении указателя мыши на определенную область страницы там появлялся бы рисунок вместо текста или значка. Динамическое воспроизведение состояния базы данных также является типичным примером ситуации, когда статическая страница не может решить задачу. Например, многие интернет-магазины поддерживают базу данных продаваемых товаров, и вывод количества оставшихся в наличии товаров требует динамического обновления соответствующего поля веб-страницы.

Веб-страницы, которые могут генерировать выводимое на экран содержание, меняющееся в зависимости от некоторых внешних условий, называются **динамическими**.

Динамика страницы достигается путем ее программирования, обычно для этого используются программные языки сценариев, такие как Perl, PHP или JavaScript.

Различают два класса программ, предназначенных для создания динамического содержания веб-страниц:

- программы, работающие на стороне клиента (то есть на том компьютере, где запущен веб-браузер, воспроизводящий страницу на экране);
- программы, работающие на стороне сервера.

[1] Об аутентификации и авторизации читайте в главе 24.

В том случае, когда программа работает на стороне клиента, код страницы передается веб-сервером веб-браузеру как обычный статический объект, а затем браузер выполняет этот код, с его помощью создает динамическое содержание страницы и выводит ее на экран. Примером может служить код, написанный на языке ActionScript, который иногда используется для программирования интерактивной анимации в играх. Однако для этого требуется еще один механизм, поддерживаемый современными браузерами, — механизм надстроек (add-on). Механизм надстроек является программным интерфейсом между браузером и внешними программами, которые расширяют функциональные возможности браузеров. Программа-надстройка обрабатывает объекты веб-страницы определенного типа, в данном случае — код ActionScript. Программой-надстройкой, которая понимает ActionScript, является Flash-плейер компании Adobe. Если Flash-плейер загружен в браузер, то динамическая веб-страница, в которой есть код ActionScript, будет правильно работать и воспроизводить интерактивную анимацию. Другим популярным языком программирования страниц на стороне клиента является JavaScript.

При программировании содержания страницы на стороне сервера процесс выглядит немного сложнее, так как программный код страницы создает содержание на сервере, следовательно, здесь нужен дополнительный этап — передача этого содержания по протоколу HTTP на клиентскую машину браузеру. Популярными языками сценариев для серверной части являются Perl, ASP, JSP и PHP. Существует также стандартный программный интерфейс между веб-сервером и программами, генерирующими динамическое содержание, — это общий шлюзовой интерфейс (Common Gateway Interface, CGI).

IP-телефония

IP-телефония — это сервис, который обеспечивает коммутируемые голосовые соединения преимущественно по схеме «один к одному» и который поддерживается сетью, использующей протокол IP в форме общедоступного Интернета или частной IP-сети.

О ТЕРМИНАХ

Понятие «IP-телефония» распространяется также и на те случаи, когда голос и факс передаются вместе с другими видами информации, в частности с текстом и изображением. Помимо термина «IP-телефония» употребляются также термины «VoIP» (Voice over IP — голос через IP) и «интернет-телефония». Хотя аббревиатура VoIP часто используется как синоним термина «IP-телефония», существует ее более широкая трактовка — любая услуга, включающая передачу голоса по протоколу IP; это может быть, например, передача голосовой рекламы при щелчке на соответствующем значке, расположенном на веб-странице. Интернет-телефония — это частный случай IP-телефонии, когда разговор происходит через Интернет, а не, например, в пределах локальной сети предприятия.

Ранняя IP-телефония

В своем развитии IP-телефония прошла три этапа.

На *первом этапе* это была, скорее, интернет-игрушка, пригодная разве что для общения двух энтузиастов, готовых мириться с сопровождающим диалог кваканьем и шипением. Два компьютера, оснащенные микрофонами, динамиками, звуковыми картами с поддержкой оцифровки звука и не очень сложным программным обеспечением, позволяли вести двусторонний диалог через Интернет в реальном времени (рис. 23.5).

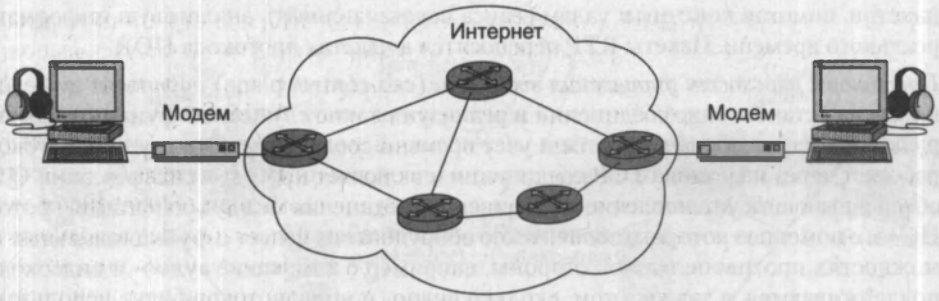


Рис. 23.5. Средства поддержки разговора пользователей через Интернет

Однако до удобств обычной телефонной услуги такой способ общения явно недотягивал. Абонентам нужно было знать IP-адрес компьютера собеседника, договариваться о времени разговора, выбирать момент для более качественной передачи речи, когда трафик Интернета между данными конкретными точками не сталкивался с перегрузками и задержками. Кроме того, при отсутствии стандартов на обоих компьютерах требовалось установить такое программное обеспечение, которое поддерживало бы один и тот же способ кодирования голоса и упаковки его в пакеты. Взаимодействия между компьютером и телефоном, подключенным к обычной телефонной сети, не предполагалось. Зато затраты ограничивались небольшой платой провайдеру за обычное коммутируемое подсоединение к Интернету.

Второй этап ознаменовался появлением стандартов IP-телефонии, прежде всего — стандартов группы H.323, разработанных ITU-T, и стандартов на основе протокола SIP, разработанного IETF.

К *третьему этапу* можно отнести появление нового поколения IP-телефонии, поддерживающей широкий спектр дополнительных услуг, подобный тому, который предоставляют абонентам развитые телефонные сети.

Стандарты H.323

Разработчики **стандартов H.323** исходили из того, что две сети — телефонная и IP — будут сосуществовать бок о бок достаточно длительное время, а значит, важно регламентировать их взаимодействие с учетом существующих в традиционных телефонных сетях процедур установления соединения, а также договориться о способе передачи вызова и собственно голоса по IP-сети.

В рамках установленного сеанса H.323 абоненты могут обмениваться не только голосовой, но и видеоинформацией, то есть пользоваться видеотелефонами или оборудованием для организации видеоконференций.

В стандартах H.323 определяется две группы протоколов (рис. 23.6):

- *Протоколы транспортной* (transport plane), или *пользовательской* (user plane), *плоскости* отвечают за непосредственную передачу голоса по сети с коммутацией пакетов. Протоколы этой плоскости определяют способы кодирования голоса (сюда входят стандарты различных кодеков, например G.711, G.723.1, G.729, G.728 и др.) и видео (кодеки H.261, H.263 и др.). Голос и видео передаются в пакетах протокола RTP (Real Time Protocol — протокол реального времени), который определен в RFC 3550 (ftp://ftp.rfc-editor.org/in-notes/rfc3550.txt) и переносит отметки времени и последовательные номера

пакетов, помогая конечным узлам сеанса восстанавливать аналоговую информацию реального времени. Пакеты RTP переносятся в пакетах протокола UDP.

- *Протоколы плоскости управления вызовами* (call control plane) переносят по сети запросы на установление соединений и реализуют такие служебные функции, как авторизация доступа абонента к сети и учет времени соединения. Эта группа протоколов работает через надежные TCP-соединения и включает протокол сигнализации Q.931, обеспечивающий установление и завершение соединения между абонентами; протокол H.245, с помощью которого абонентское оборудование узнает о функциональных возможностях противоположной стороны, например о том, какие аудио- и видеокодеки поддерживаются, а также о том, сколько аудио- и видеопотоков будут использовать абоненты в рамках данного соединения. По умолчанию IP-телефон поддерживает только один голосовой поток, но видеотелефон уже поддерживает два потока — один голосовой и один видео, а оборудование видеоконференци может поддерживать несколько аудиопотоков и несколько видеопотоков. Еще один протокол этой группы — RAS (Registration, Admission, Status) — служит для учета звонков, регистрации пользователя в некотором административном домене (например, в домене организации, где работает пользователь) и контроля доступа в сеть (то есть проверке сетевых ресурсов, таких как свободная пропускная способность, необходимых для качественного обслуживания телефонного вызова).

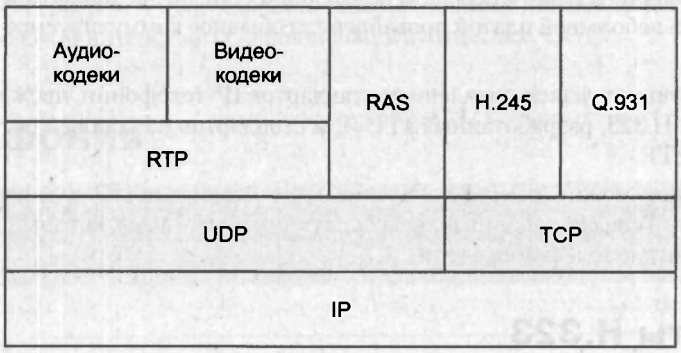


Рис. 23.6. Стек протоколов H.323

Основными элементами сети H.323, в которых реализуются протоколы этого стека, являются так называемые IP-телефоны, подключаемые непосредственно к IP-сеги, и шлюзы, связывающие традиционную телефонную сеть с IP-сетью (рис. 23.7).

Шлюз (gateway) обеспечивает трансляцию упакованного в пакеты оцифрованного и зачастую сжатого голоса в форму, пригодную для передачи по телефонной сети общего пользования. Кроме того, в функции шлюза H.323 входит трансляция протоколов сигнализации телефонных сетей, таких, например, как SS7, в протоколы сигнализации стека H.323. Шлюз позволяет абонентам с обычным телефонным аппаратом общаться с пользователями IP-телефонов или же задействовать IP-сеть как транзитную.

Основная задача плоскости управления вызовами — установление соединения между абонентами через сети с коммутацией пакетов — в простейшем случае может быть решена шлюзом, а в более общей постановке поручается специальному элементу сети — привратнику.

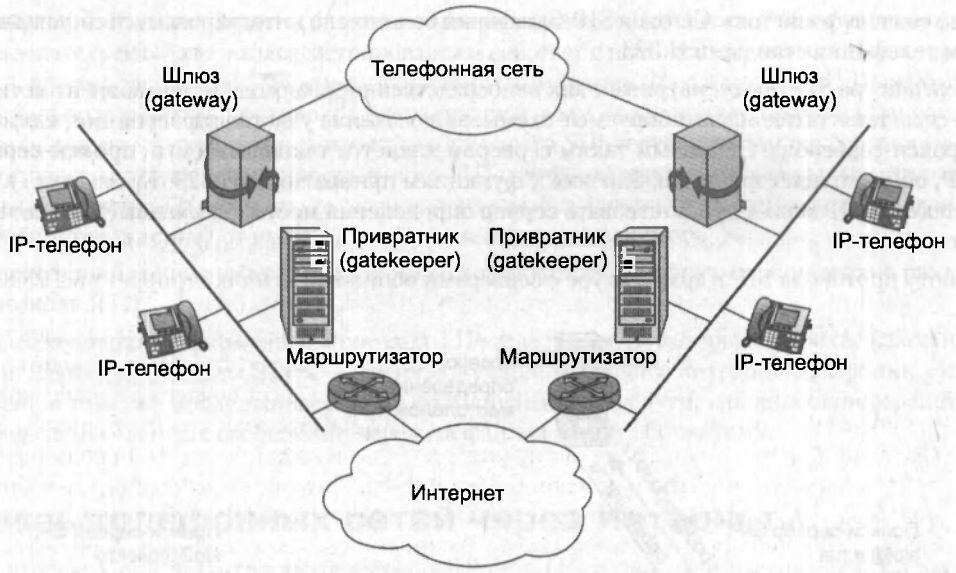


Рис. 23.7. Элементы сети H.323

Привратник (gatekeeper) выполняет регистрацию и авторизацию абонентов по протоколу RAS, а также, в случае необходимости, трансляцию адресов (например, DNS-имен в телефонные номера). Кроме того, он занимается маршрутизацией вызовов к IP-телефону или шлюзу, а если потребуется, то и к другому привратнику.

Обычно один привратник обслуживает так называемую зону, то есть часть сети, находящуюся под административным управлением одной организации. Все функции привратника в архитектуре H.323 могут выполнять терминальные устройства — телефоны и шлюзы, но такое решение плохо масштабируется, а поток вызовов с трудом контролируется и тарифицируется.

Стандарты на основе протокола SIP

Основным конкурентом протоколов стандарта H.323 является протокол **SIP** (Session Initiation Protocol — протокол инициирования сеанса), разработанный интернет-сообществом и стандартизованный IETF в RFC 3261 (ftp://ftp.rfc-editor.org/in-notes/rfc3261.txt).

SIP является протоколом сигнализации, он ответственен за установление сеанса между абонентами, при этом SIP выполняет функции протоколов Q.931, RAS и H.245 стандарта H.323 (точнее — часть из них). Для передачи аудио- и видеоданных в ходе сеанса протокол SIP предполагает использование протокола RTP.

Протокол SIP очень близок по стилю к протоколу HTTP: он имеет похожий набор и синтаксис сообщений, которыми обмениваются стороны в процессе установления сеанса. Как и у протокола HTTP, SIP-сообщения текстовые, они хорошо понятны программистам, имеющим опыт создания веб-приложений. Поэтому системы IP-телефонии, построенные на основе SIP, оказались гораздо ближе к миру Интернета, чем стандарты H.323, пришед-

шие «от телефонистов». Сегодня SIP-телефония более тесно интегрирована с веб-услугами, чем телефония стандарта H.323.

Архитектура SIP предусматривает как непосредственное взаимодействие абонентов через IP-сеть, так и более масштабируемые схемы, включающие участие серверов-посредников (прокси-серверов). Основным таким сервером является так называемый **прокси-сервер SIP**, он выполняет функции, близкие к функциям привратника H.323. Кроме того, в архитектуре SIP может присутствовать **сервер определения местоположения** (SIP Location Server).

Работу протокола SIP в архитектуре с серверами обоих типов иллюстрирует рис. 23.8.

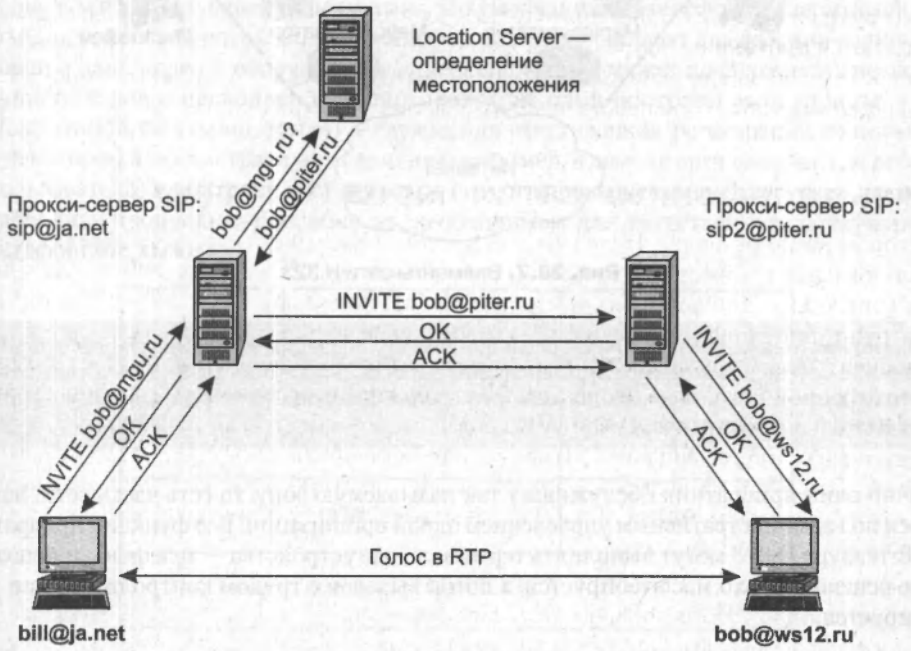


Рис. 23.8. Взаимодействие абонентов SIP

Адресами абонентов в протоколе SIP являются универсальные идентификаторы URI, используемые во всех веб-службах. На рис. 23.8 абонент bill@ja.net хочет установить сеанс с абонентом bob@mgu.ru. В домене ja.net установлен прокси-сервер SIP с именем sip1@ja.net, через него проходят все вызовы абонентов этого домена (за счет того, что в IP-телефонах абонентов задан IP-адрес этого прокси-сервера).

Запросом на установление сеанса в протоколе SIP является передача сообщения *INVITE* с URI вызываемого абонента, поэтому абонент bill@ja.net направляет своему прокси-серверу сообщение *INVITE bob@mgu.ru*. Прокси-сервер для выполнения этого запроса обращается к серверу определения местоположения, который возвращает ему ответ о том, что абонент bob@mgu.ru в данный момент зарегистрирован как активный в домене piter.ru с именем bob@piter.ru. Прокси-сервер использует эту информацию для того, чтобы направить сообщение *INVITE* прокси-серверу домена piter.ru (сервер с именем sip2@piter.ru), указав в нем имя

bob@piter.ru. Вызов завершается прокси-сервером sip2@piter.ru, который обнаруживает, что пользователь bob@piter.ru зарегистрировался и работает в настоящее время за компьютером ws12, поэтому вызов *INVITE* передается на этот компьютер. Далее протокол SIP работает подобно большинству протоколов сигнализации: если пользователь bob@ws12.ru соглашается принять вызов, то он снимает трубку своего SIP-телефона (или щелкает на соответствующем значке своего программного SIP-телефона) и тем самым посылает ответ *OK* назад по цепочке. Окончательное установление сеанса фиксируется отправкой сообщения *ACK* (подтверждение) от вызывающего абонента к вызываемому.

После установления сеанса разговор происходит между телефонами абонентов в рамках протокола RTP.

Существуют также фирменные протоколы IP-телефонии, из которых наиболее известными являются **протоколы Skype** — очень популярного сервиса интернет-телефонии. Этот сервис к тому же поддерживает такие дополнительные услуги, как видеоконференции, передача мгновенных сообщений, передача файлов между абонентами.

Связь телефонных сетей через Интернет

На втором этапе развития IP-телефонии IP-сеть (Интернет или частная сеть) широко использовалась в качестве транзитной сети между двумя местными телефонными сетями (рис. 23.9). Данная схема реализации общедоступных услуг IP-телефонии стала достаточно популярной во всем мире, в том числе в России. Она заключается в том, что абонент звонит по определенному номеру, который закреплен за провайдером местной телефонной сети, и на звонок отвечает **сервер интерактивного голосового ответа** (Interactive Voice Response, IVR). IVR-сервер запрограммирован на выполнение рутинных процедур аутентификации вызывающего абонента и приема номера вызываемого абонента. Для этого привлекается техника распознавания голосовых ответов (которыми могут быть и сигналы тонового набора, используемого вызывающим абонентом для ответов на запросы IVR-сервера).

Для реализации услуги IP-телефонии по описанной схеме оператору связи не надо создавать собственную дорогостоящую транспортную инфраструктуру и иметь непосредственный доступ к абонентам. Однако стратегические перспективы такого подхода оставляют желать лучшего из-за плохой масштабируемости и узкого спектра услуг.

Масштабируемость такого варианта ограничивается несколькими факторами. Во-первых, провайдеру приходится устанавливать многочисленные одноранговые связи со своими друзьями-соперниками по бизнесу. Во-вторых, протоколы обеих плоскостей необходимо реализовывать во всех элементах сети IP-телефонии: и в привратниках, и в шлюзах, и в терминалах, что приводит к излишней сложности и дороговизне всех этих устройств. И наконец, пользователям предоставляются только базовые услуги по обработке вызовов, поскольку взаимодействие с протоколами межстанционной сигнализации (SS7) и службами интеллектуальной сети (IN) отсутствует. Эту последнюю группу недостатков нельзя отнести на счет стандартов H.323, в которых явно не говорится о том, какие протоколы сигнализации должен поддерживать шлюз со стороны телефонной сети. Перечень дополнительных услуг по обработке вызовов определен в спецификации H.450. Таким образом, это скорее изъян реализации шлюзов того поколения, в которых поддержка SS7 и IN, как правило, отсутствовала.

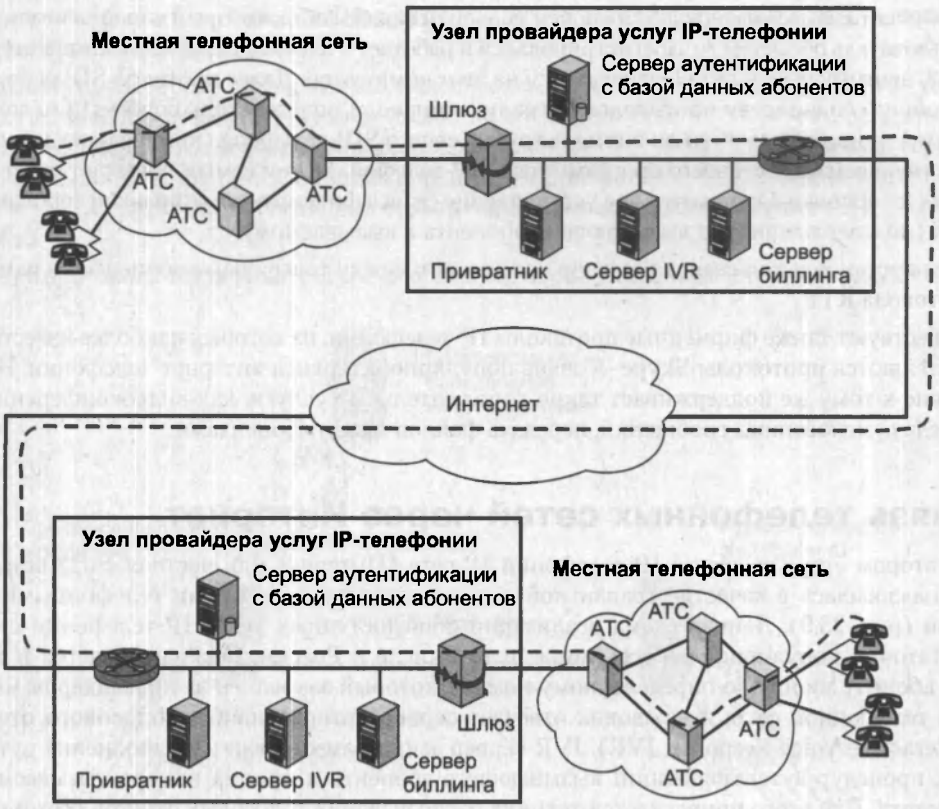


Рис. 23.9. Взаимодействие двух местных телефонных сетей через Интернет

Кроме того, сам диалог достаточно утомителен — гораздо удобнее просто набрать номер с небольшой приставкой вроде 8-20 и получить доступ к услугам международной IP-телефонии. Но для этого провайдеру нужен прямой доступ к абоненту или договоренность с местными операторами о переадресации таких вызовов на шлюз IP-телефонии провайдера с помощью средств интеллектуальной сети (а они пока поддерживаются далеко не всеми местными операторами). Таким образом, для выхода IP-телефонии на более высокий уровень национального или международного оператора требуются другие стандарты и оборудование, чтобы сети, построенные на базе протокола IP, могли равноправно соседствовать с традиционными телефонными сетями.

Многие из необходимых стандартов уже появились и воплощены в новом поколении оборудования, ставшим основой для *третьего этапа* развития IP-телефонии.

Новое поколение сетей IP-телефонии

Укрупненная схема полномасштабной сети IP-телефонии показана на рис. 23.10. Такая сеть может поддерживать собственных абонентов и служить транзитной для традиционных телефонных сетей с оказанием полного спектра услуг, включая услуги интеллектуальной сети.

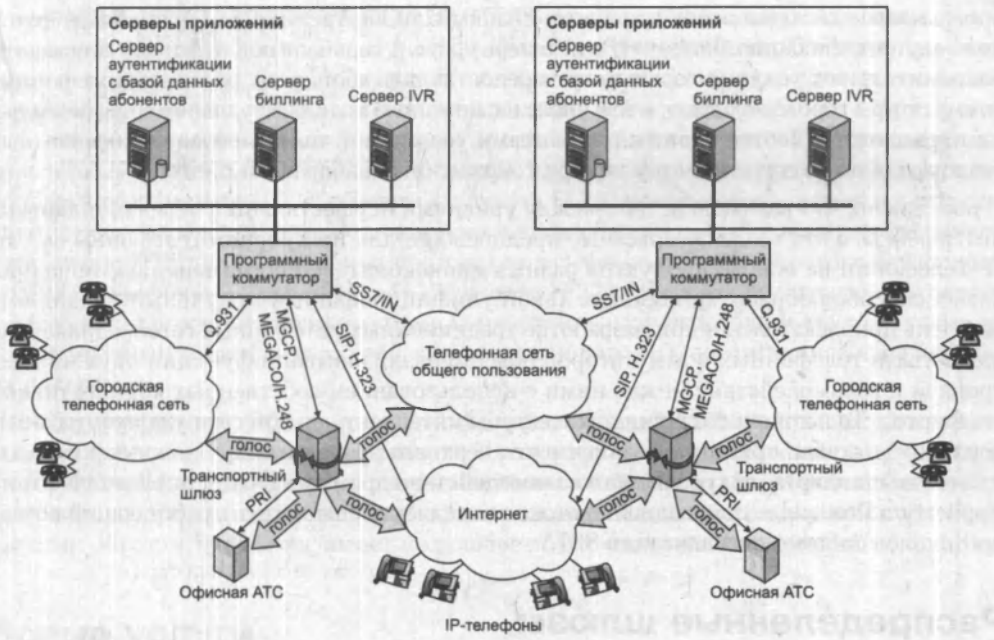


Рис. 23.10. Масштабируемая архитектура IP-телефонии

Эта сеть обладает несколькими отличительными особенностями. Так, в узлах IP-телефонии нового поколения произошло четкое разделение функций на три группы:

- транспортную;
- управления вызовами;
- прикладных сервисов.

Транспортная группа образовалась за счет выделения из шлюза функциональной части, выполняющей очень простую операцию — коммутацию между входными и выходными портами (физическими или виртуальными). Этот элемент, получивший название **транспортного шлюза** (Media Gateway, MG), является своего рода аналогом коммутационного поля телефонной станции.

Следующую группу — группу управления вызовами — составляют протоколы сигнализации IP-телефонии (H.225.0, RAS из стандарта H.323 или SIP). К этой группе относят также протоколы управления транспортными шлюзами, которые инициируют действия по коммутации портов. Все перечисленные базовые функции по обработке вызовов сегодня часто реализуются одним устройством — так называемым **программным коммутатором** (softswitch).

Третья группа функций образует уровень сервисов, реализуемых в виде обычных сетевых приложений универсальными серверами. Примерами таких сервисов являются инициирование телефонного вызова при щелчке на определенной кнопке веб-страницы, передача вызова абоненту, подключенному к Интернету по телефонной сети, а также услуги интеллектуальной сети.

В сетях IP-телефонии второго этапа развития уровень сервисов практически отсутствовал — пользовательские услуги оказывал только IVR-сервер, а остальные прикладные

программные системы этого уровня реализовывали внутренние для провайдера функции — аутентификацию, биллинг и т. п. Теперь уровень сервисов поддерживает весь спектр дополнительных услуг, которые могут предоставлять абонентам развитые телефонные коммутаторы городского типа, в том числе и с помощью интеллектуальной сети: переадресацию вызовов в соответствии с различными условиями, телеголосование, бесплатный звонок, звонок по специальному тарифу, сокращенный набор и т. п.

Очень важно, что взаимодействие между уровнями осуществляется через стандартные интерфейсы, а это создает серьезные предпосылки для построения телефонных узлов IP-телефонии на основе продуктов разных производителей с применением общепринятых способов обработки вызовов. Такой унифицированный модульный подход был бы очень привлекателен и при разработке традиционных телефонных сетей, однако производители телефонных коммутаторов обычно реализовывали функции двух нижних уровней и взаимодействие между ними с использованием собственных корпоративных стандартов. Только при создании архитектуры интеллектуальной сети удалось, наконец, воплотить в жизнь принцип независимости верхнего уровня от двух нижних и принять в качестве стандарта межуровневого взаимодействия протокол **INAP** (Intelligent Network Application Protocol — прикладной протокол интеллектуальной сети), работающий поверх протоколов системы сигнализации SS7.

Распределенные шлюзы и программные коммутаторы

Масштабируемость коммутации и независимость транспортного уровня от уровня управления вызовами в новом поколении узлов IP-телефонии достигается благодаря применению концепции программного коммутатора. Сам термин «softswitch» получил широкое распространение в названиях продуктов, компаний и неформальных объединений. Ни в одном из современных стандартов нет определения программного коммутатора, но этот маркетинговый термин выделяет в архитектуре распределенного узла IP-телефонии некоторый общий элемент. Данный управляющий элемент отвечает за обработку сообщений протоколов сигнализации, на основании которых происходят соединения: например, протокола H.225.0 стека H.323, протокола установления соединений SIP или же протокола сигнализации SS7.

С помощью специального протокола «главный-подчиненный» программный коммутатор управляет транспортными шлюзами, которые, в конечном счете, и осуществляют коммутацию голосовых каналов. Для управления шлюзами сегодня могут использоваться несколько близких по логике работы протоколов: **SGCP** (Simple Gateway Control Protocol), **MGCP** (Media Gateway Control Protocol) или **MEGACO/H.248**. Собственно, стандартом, принятым как IETF, так и ITU-T, является только совместно разработанный ими протокол MEGACO/H.248, однако и предшественники этого стандарта, протоколы SGCP и MGCP, успешно реализуются в продуктах различных производителей. С помощью одного из названных протоколов программный коммутатор выясняет детали текущего состояния соединений и портов шлюза, а также передает ему указания о том, какую пару портов (физических или логических) требуется соединить, и некоторые другие предписания. Таким образом, реализация шлюза может быть весьма простой, а весь интеллект управления соединениями перемещается на уровень программного коммутатора, который в модели распределенной коммутации управляет одновременно несколькими шлюзами. Именно такой вариант показан на рис. 23.10.

В протоколах SGCP, MGCP и MEGACO/H.248 управляющий элемент называется **агентом вызова** (call agent), однако программный коммутатор — это нечто большее, чем агент управления вызовами. Обычно в продукт с маркой softswitch производители помещают элементы уровня управления вызовами нескольких стандартов, чтобы такой программный коммутатор мог взаимодействовать с другими зонами телефонной сети по наиболее популярным протоколам сигнализации. Так, в программный коммутатор может входить привратник стандарта H.323, серверы стандарта SIP (прокси-сервер, сервер переадресации и сервер определения местоположения пользователей), а также шлюзы телефонной сигнализации для преобразования протоколов телефонных сетей в протоколы сигнализации IP-телефонии — те же SIP и H.225.0 стека H.323. Широкая поддержка протоколов сигнализации позволяет программному коммутатору находить общий язык практически с любыми типами телефонных сетей, как с традиционными (с коммутацией каналов), так и с пакетными.

Программные коммутаторы — «сердце» современного узла IP-телефонии — осуществляют за единицу времени множество соединений, столько же, сколько телефонные коммутаторы городского и междугородного типов. Высокая степень масштабируемости достигается благодаря распределенной модели коммутации, элементы которой взаимодействуют стандартным образом, что обеспечивает модульное построение узла коммутации.

Новые услуги

В промежуточных устройствах IP-сети не хранится информация о каждом соединении абонентов (компьютеров пользователей) с серверами. Это одно из принципиальных отличий IP-сети от телефонной сети. Коммутаторы телефонной сети, напротив, отслеживают и запоминают состояние каждого вызова, что является одной из причин более высокой стоимости передачи через них транзитного трафика по сравнению с IP-маршрутизаторами.

В публикациях по IP-телефонии постоянно подчеркивается, что удешевление звонков и оказание конкурентного давления на сектор традиционной международной телефонии — это краткосрочное преимущество IP-телефонии. Что же касается дальней стратегической перспективы, то основным направлением здесь будет предоставление новых услуг, в том числе интегрированных с услугами по передаче данных и манипулированию данными. К ним относятся:

- Click to Talk — инициирование телефонного разговора при просмотре веб-страницы Web;
- Internet Call Waiting (ICW) — уведомление абонента, подключившегося с помощью телефонной сети к Интернету, о наличии входящего вызова и, возможно, организация параллельного с интернет-сеансом разговора путем пакетной передачи;
- Unified Messaging — организация единой почтовой службы для любых сообщений, в том числе электронной почты, факсов и голоса, с возможностью трансформации вида представления информации.

Разнообразие услуг, их настройка в соответствии с потребностями конкретного пользователя, простота программирования нового предложения, легкость интеграции голосовых услуг с услугами манипулирования данными — это «врожденные» сильные стороны IP-телефонии, ее стратегический потенциал. Часть этих услуг, описываемых стандартами SIP и H.245 как дополнительные, может предоставлять непосредственно программный коммутатор, более сложные сервисы реализуются с помощью серверов приложений узла IP-телефонии.

Интеграция систем адресации E.164 и DNS на основе ENUM

Одной из проблем современной IP-телефонии является сложность установления соединения, когда инициировавший вызов абонент использует обычный телефонный аппарат, подключенный к традиционной телефонной сети, а вызываемый абонент — компьютер или IP-телефон, соединенный с Интернетом или частной IP-сетью. Сложность подобного соединения связана с применением в общедоступных телефонных сетях и Интернете *разных схем адресации* — системы телефонных номеров на основе международного стандарта E.164 и системы имен DNS. И если пользователю компьютера или цифрового IP-телефона не составляет труда набрать телефонный номер для вызова абонента, то представить себе набор DNS-имени с помощью обычного аналогового аппарата довольно сложно.

Для преодоления пропасти между этими видами общедоступных услуг необходимо либо выбрать единую схему идентификации абонентов, либо разработать метод трансляции одной схемы в другую. Предложения **ENUM** (E.164 NUmber Mapping — отображение адресов стандарта E.164) рабочей группы IETF решают задачу вторым способом, и пока этот вариант наиболее близок к немедленной реализации. Подход ENUM, описанный в RFC 3761 (ftp://ftp.rfc-editor.org/in-notes/rfc3761.txt), состоит в назначении всем абонентам IP-телефонии, подключенным к Интернету или частной IP-сети, идентификаторов еще одного типа — телефонных номеров стандарта E.164. Однако на конечных узлах и даже сетях, в которых вызов терминируется, эти телефонные номера не используются — они нужны только для идентификации вызываемого абонента стороной-инициатором, применяющей обычный телефон, и маршрутизации вызова в пределах традиционной телефонной сети. Затем телефонные номера преобразуются в имена Интернета с помощью хорошо известной и отлично зарекомендовавшей себя службы — системы доменных имен (DNS).

Используемый при этом подход подобен тому, который применяется для решения обратной задачи — нахождению имени узла по его IP-адресу. С этой целью предлагается создать новую зону e164.arpa, куда будут входить территории, соответствующие цифрам телефонного номера, например, зоны верхнего уровня 1, 7, 33, 44 для номеров, принадлежащих абонентам Североамериканского региона, России, Франции и Великобритании соответственно. Домен верхнего уровня arpa традиционно отводится для решения обратной задачи — нахождение имени по адресу с помощью зоны in-addr.arpa.

Для преобразования телефонного номера в DNS-имя используется специальный тип записи — Naming Athority Pointer (NAPTR). Изначально данная запись предназначалась для перечисления сервисов, которые поддерживает организация, администрирующая данный домен (RFC 2915). Примером такой записи может служить строка sip:Petrov@firma.ru, сообщающая о том, что с абонентом можно связаться, направив ему вызов по протоколу SIP на имя Petrov@firma.ru. Очевидно, что такие записи будут находиться только в зонах самого нижнего уровня, где располагается база номеров, которую провайдер получил для обслуживания конечных абонентов. Зоны же верхнего уровня будут содержать только обычные ссылки на серверы имен зон более низкого уровня. Итак, если имени Petrov@firma.ru соответствует телефонный номер +7 095 758 35 22, то связанная с этим абонентом запись, возможно, содержится в зоне 8.5.7.5.9.0.7.e164.arpa (обратный порядок записи цифр телефонного номера согласуется с принятым в DNS правилом расположения старшей части имени справа, а не слева, как в телефонии). Запись может находиться и в зоне 3.8.5.7.5.9.0.7.e164.arpa, если все номера диапазона +7 095 758 3*xx xx* переданы еще более мелкому провайдеру (в предыдущем примере предполагалось, что все номера +7 095 758 *xx xx* принадлежали

одному провайдеру). Деление телефонного номера на зоны производится по цифрам в полном соответствии с административной ответственностью каждой конкретной организации за отображение телефонных номеров на DNS-имена (точнее, на URL-адреса, которые в дополнение к DNS-имени имеют префикс, указывающий на протокол доступа к ресурсу). Чем больше уровней подчиненности провайдеров IP-телефонии, тем больше составных компонентов в имени зоны.

Протокол передачи файлов

До появления службы WWW сетевая файловая служба на основе протокола **FTP** (File Transfer Protocol — протокол передачи файлов), описанная в спецификации RFC 959, долгое время была самой популярной службой доступа к удаленным данным в Интернете и корпоративных IP-сетях. FTP-серверы и FTP-клиенты имеются практически в каждой ОС, кроме того, для доступа ко все еще популярным FTP-архивам используются FTP-клиенты, встроенные в браузеры.

Протокол FTP позволяет целиком переместить файл с удаленного компьютера на локальный, и наоборот. FTP также поддерживает несколько команд просмотра удаленного каталога и перемещения по каталогам удаленной файловой системы. Поэтому FTP особенно удобно использовать для доступа к тем файлам, данные которых нет смысла просматривать удаленно, а гораздо эффективней целиком переместить на клиентский компьютер (например, файлы исполняемых модулей приложений).

В протокол FTP встроены примитивные средства авторизации удаленных пользователей на основе передачи по сети пароля в открытом виде. Кроме того, поддерживается анонимный доступ, не требующий указания имени пользователя и пароля; такой способ доступа часто рассматривается как более безопасный, так как он не подвергает пароли пользователей угрозе перехвата.

Основные модули службы FTP

FTP-клиент состоит из трех основных функциональных модулей.

- User Interface (аналог агента пользователя) — пользовательский интерфейс, принимающий от пользователя команды и отображающий состояние FTP-сеанса на экране. Пользовательский интерфейс зависит от программной реализации FTP-клиента. Наряду с традиционными клиентами, работающими в символьном режиме, имеются и графические оболочки, не требующие от пользователя знания символьных команд. Символьные клиенты обычно поддерживают следующий основной набор команд:
  - `open` *имя_хоста* — открытие сеанса с удаленным сервером;
  - `bye` — завершение сеанса с удаленным хостом и завершение работы утилиты ftp;
  - `close` — завершение сеанса с удаленным хостом, утилита ftp продолжает работать;
  - `ls` (`dir`) — печать содержимого текущего удаленного каталога;
  - `get` *имя_файла* — копирование удаленного файла на локальный хост;
  - `put` *имя_файла* — копирование удаленного файла на удаленный сервер.
- User-PI — интерпретатор команд пользователя. Этот модуль взаимодействует с модулем Server-PI FTP-сервера.

- User-DTP — модуль, осуществляющий передачу данных файла по командам, получаемым от модуля User-PI по протоколу клиент-сервер. Этот модуль взаимодействует с локальной файловой системой клиента.

FTP-сервер включает два модуля.

- Server-PI — модуль, который принимает и интерпретирует команды, передаваемые по сети модулем User-PI.
- Server-DTP — модуль, управляющий передачей данных файла по командам от модуля Server-PI. Взаимодействует с локальной файловой системой сервера.

Управляющий сеанс и сеанс передачи данных

FTP-клиент и FTP-сервер поддерживают параллельно два сеанса — управляющий сеанс и сеанс передачи данных. *Управляющий сеанс* открывается при установлении первоначального FTP-соединения клиента с сервером, причем в течение одного управляющего сеанса может последовательно выполняться несколько *сеансов передачи данных*, в рамках которых передаются или принимаются несколько файлов.

Общая схема взаимодействия клиента и сервера выглядит следующим образом.

1. FTP-сервер всегда открывает управляющий TCP-порт 21 для прослушивания, ожидая прихода запроса на установление управляющего FTP-соединения от удаленного клиента.
2. После установления управляющего соединения FTP-клиент отправляет на сервер команды, которые уточняют параметры соединения: имя и пароль клиента, роль участников соединения (активная или пассивная), порт передачи данных, тип передачи, тип передаваемых данных (двоичные данные или код ASCII), директивы на выполнение действий (читать файл, писать файл, удалить файл и т. п.).
3. После согласования параметров пассивный участник соединения переходит в режим ожидания открытия соединения на порт передачи данных. Активный участник инициирует это соединение и начинает передачу данных.
4. После окончания передачи данных соединение по портам данных закрывается, а управляющее соединение остается открытым. Пользователь может по управляющему соединению активизировать новый сеанс передачи данных.

Порты передачи данных выбирает FTP-клиент (по умолчанию клиент может использовать для передачи данных порт управляющего сеанса), а сервер должен задействовать порт, номер которого на единицу меньше номера порта клиента.

Команды взаимодействия FTP-клиента с FTP-сервером

В протоколе FTP предусмотрены специальные команды для взаимодействия FTP-клиента с FTP-сервером (не следует их путать с командами пользовательского интерфейса клиента, ориентированные на применение человеком). Эти команды делятся на три группы.

- *Команды управления доступом к системе* доставляют серверу имя и пароль клиента, изменяют текущий каталог на сервере, повторно инициализируют, а также завершают управляющий сеанс.

- *Команды управления потоком данных* устанавливают параметры передачи данных. Служба FTP может применяться для передачи разных типов данных (код ASCII или двоичные данные), работать как со структурированными данными (файл, запись, страница), так и с неструктурированными.
- *Команды службы FTP* управляют передачей файлов, операциями над удаленными файлами и каталогами. Например, команды `RETR` и `STOR` запрашивают передачу файла соответственно от сервера на клиентский хост, и наоборот. Параметрами каждой из этих команд является имя файла. Может быть задано также смещение от начала файла — это позволяет начать передачу файла с определенного места при непредвиденном разрыве соединения. Команды `DELE`, `MKD`, `RMD`, `LIST` соответственно удаляют файл, создают каталог, удаляют каталог и передают список файлов текущего каталога. Каждая команда протокола FTP передается в виде одной строки кода ASCII.

Сетевое управление в IP-сетях

Функции систем управления

Любая сложная вычислительная сеть требует дополнительных специальных средств управления помимо имеющихся в стандартных сетевых операционных системах. Это связано с большим количеством разнообразного коммуникационного оборудования, работа которого критически важна для выполнения сетью своих основных функций.

Распределенный характер крупной сети делает невозможным поддержание ее работы без централизованной **системы управления сетью** (Network Management System, NMS), призванной в автоматическом режиме контролировать сетевой трафик и управлять коммуникационным оборудованием сети.

Системы управления сетью работают, как правило, в *автоматизированном* режиме, выполняя наиболее простые действия автоматически и оставляя сложные решения для принятия человеку на основе подготовленной системой информации.

Системы управления сетью представляют собой сложные программно-аппаратные комплексы, поэтому существует граница целесообразности их применения. В небольшой сети можно применять отдельные программы управления наиболее сложными устройствами, например коммутатором, поддерживающим технику VLAN. Обычно каждое устройство, которое требует достаточно сложного конфигурирования, производитель сопровождает автономной программой конфигурирования и управления. Однако при росте сети может возникнуть проблема объединения разрозненных программ управления устройствами в единую систему управления, и для решения этой проблемы придется, возможно, отказаться от этих программ и заменить их интегрированной системой управления сетью.

В соответствии с рекомендациями ITU-T X.700 и стандарта ISO 7498-4 система управления сетью должна решать следующие группы задач:

- **Управление конфигурацией сети и именованием** заключаются в конфигурировании параметров как **элементов сети** (Network Element, **NE**), так и сети в целом. Для элементов сети, таких как маршрутизаторы, мультиплексоры и т. п., путем конфигурирования определяются сетевые адреса, идентификаторы (имена), географическое положение и пр. Для сети в целом управление конфигурацией обычно начинается с построения карты сети, то есть с отображения реальных связей между элементами сети и связей

между элементами сети, иллюстрирующих образование новых физических или логических каналов, изменение таблиц коммутации и маршрутизации.

- **Обработка ошибок** включает выявление, определение и устранение последствий сбоев и отказов в работе сети.
- **Анализ производительности и надежности** связан с оценкой на основе накопленной статистической информации таких параметров, как время реакции системы, пропускная способность реального или виртуального канала связи между двумя конечными абонентами сети, интенсивность трафика в отдельных сегментах и каналах сети, вероятность искажения данных при их передаче через сеть, а также коэффициент готовности сети или ее определенной транспортной службы. Результаты анализа производительности и надежности позволяют контролировать *соглашение об уровне обслуживания* (SLA), заключаемое между пользователем сети и ее администраторами (или компанией, продающей услуги). Без средств анализа производительности и надежности поставщик услуг публичной сети или отдел информационных технологий предприятия не сможет ни проконтролировать, ни тем более обеспечить нужный уровень обслуживания для конечных пользователей сети.
- **Управление безопасностью** подразумевает контроль доступа к ресурсам сети (данным и оборудованию) и сохранение целостности данных при их хранении и передаче через сеть. Базовыми элементами управления безопасностью являются процедуры аутентификации пользователей, назначение и проверка прав доступа к ресурсам сети, распределение и поддержка ключей шифрования, управления полномочиями и т. п. Часто функции этой группы не включаются в системы управления сетями, а либо реализуются в виде специальных продуктов обеспечения безопасности, например сетевых экранов или централизованных систем авторизации[1], либо входят в состав операционных систем и системных приложений.
- **Учет работы сети** включает регистрацию времени использования различных ресурсов сети (устройств, каналов и транспортных служб) и ведение биллинговых операций (плата за ресурсы). Ввиду специфического характера оплаты услуг у различных поставщиков и различными формами соглашения об уровне обслуживания, эта группа функций реализуется только в нестандартных системах, разрабатываемых для конкретного заказчика.

В стандартах, определяющих перечисленные функции систем управления, не делается различий между управляемыми объектами — каналами, сегментами локальных сетей, коммутаторами и маршрутизаторами, модемами и мультиплексорами, аппаратным и программным обеспечением компьютеров, однако на практике деление систем управления по типам управляемых объектов широко распространено.

Ставшими классическими системы управления сетями, такие как SunNet Manager, HP OpenView или Cabletron Spectrum, управляют только *коммуникационными* объектами корпоративных сетей, такими как маршрутизаторы и коммутаторы.

В тех случаях, когда управляемыми объектами являются *компьютеры*, а также их системное и прикладное программное обеспечение, то для системы управления часто используют особое название — **система управления системой** (System Management System, SMS).

[1] О средствах обеспечения сетевой безопасности читайте в главе 24.

SMS обычно автоматически собирает информацию об установленных в сети компьютерах и создает записи в специальной БД об аппаратных и программных ресурсах. SMS может централизованно устанавливать и администрировать приложения, которые запускаются с файловых серверов, а также удаленно измерять наиболее важные параметры компьютера, операционной системы, СУБД (например, коэффициент использования процессора или физической памяти, интенсивность страничных прерываний и др.). SMS может давать администратору возможность брать на себя удаленное управление компьютером в режиме эмуляции графического интерфейса популярных операционных систем.

Заметим, что в последние годы существует отчетливая тенденция интеграции систем управления сетями и систем управления системами.

Архитектуры систем управления сетями

Основным элементом любой системы управления сетью является схема взаимодействия «менеджер — агент — управляемый объект» (рис. 23.11). На основе этой схемы могут быть построены системы практически любой сложности с большим количеством агентов, менеджеров и ресурсов разного типа.

Рис. 23.11. Взаимодействие агента, менеджера и управляемого объекта

Чтобы можно было автоматизировать управление объектами сети, создается некоторая *модель* управляемого объекта, называемая **базой данных управляющей информации** (Management Information Base, MIB). MIB отражает только те характеристики объекта, которые нужны для его контроля. Например, модель маршрутизатора обычно включает такие характеристики, как количество портов, их тип, таблицу маршрутизации, количество кадров и пакетов протоколов канального, сетевого и транспортного уровней, прошедших через эти порты.

Менеджер и агент работают с одной и той же моделью управляемого объекта, однако в использовании этой модели агентом и менеджером имеются существенные различия.

Агент наполняет MIB управляемого объекта текущими значениями его характеристик, а **менеджер** извлекает из MIB данные, на основании которых он узнает, какие характеристики он может запросить у агента и какими параметрами объекта можно управлять. Таким образом, агент является посредником между управляемым объектом и менеджером. Агент поставляет менеджеру только те данные, которые предусматриваются MIB.

Менеджер и агент взаимодействуют по стандартному протоколу. Этот протокол позволяет менеджеру запрашивать значения параметров, хранящихся в MIB, а также передавать агенту информацию, на основе которой тот должен управлять объектом. Обычно менеджер работает на отдельном компьютере, взаимодействуя с несколькими агентами.

Агенты могут встраиваться в управляемое оборудование или работать на отдельном компьютере, связанном с управляемым оборудованием. Для получения требуемых данных об объекте, а также для выдачи на него управляющих воздействий агент должен иметь возможность взаимодействовать с ним. Однако многообразие типов управляемых объектов не позволяет стандартизовать способ взаимодействия агента с объектом. Эта задача решается разработчиками при встраивании агентов в коммуникационное оборудование или в операционную систему. Агент может снабжаться специальными датчиками для получения информации, например датчиками релейных контактов или датчиками температуры. Агенты могут отличаться разным уровнем интеллекта: обладать как самым минимальным интеллектом, необходимым для подсчета проходящих через оборудование кадров и пакетов, так и весьма высоким, достаточным для самостоятельных действий по выполнению последовательности управляющих команд в аварийных ситуациях, построению временны́х зависимостей, фильтрации аварийных сообщений и т. п.

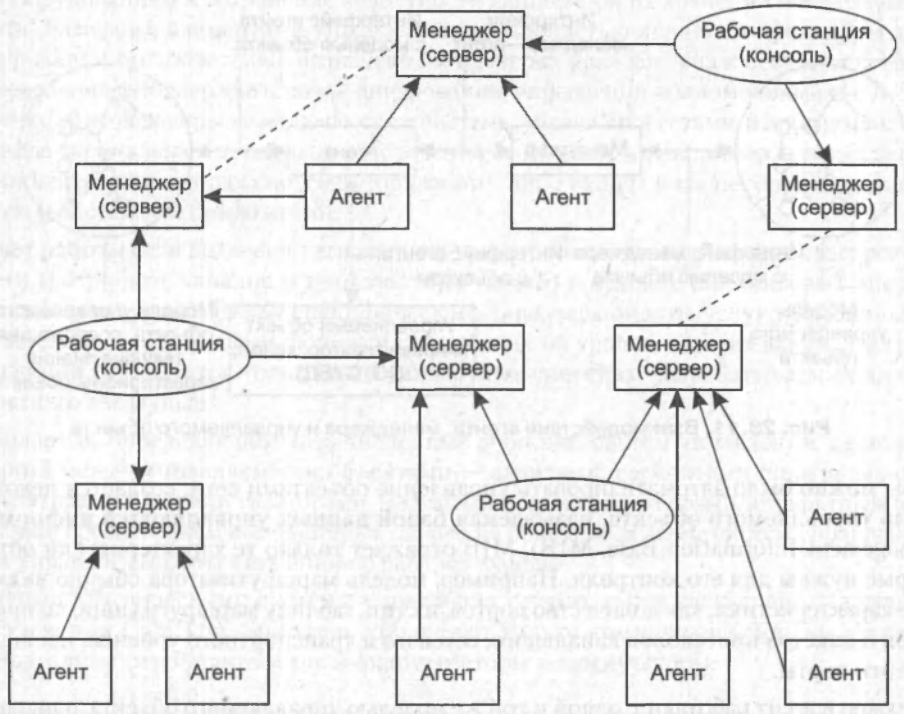


Рис. 23.12. Распределенная система управления на основе нескольких менеджеров и рабочих станций

Различают *внутриполосное управление*, когда управляющие сигналы идут по тому же каналу, по которому передаются пользовательские данные, и *внеполосное управление*, то есть осуществляемое вне канала, по которому передаются пользовательские данные. Внутриполосное управление более экономично, так как не требует создания отдельной инфраструктуры передачи управляющих данных. Однако внеполосное управление на-

дежнее, так как соответствующее оборудование может выполнять свои функции даже тогда, когда те или иные сетевые элементы выходят из строя, и основные каналы передачи данных оказываются недоступными.

Схема «менеджер — агент — управляемый объект» позволяет строить достаточно сложные в структурном отношении распределенные системы управления (рис. 23.12).

Каждый агент, показанный на рисунке, управляет одним или несколькими элементами сети, параметры которых он помещает в соответствующую базу MIB. Менеджеры извлекают данные из баз MIB своих агентов, обрабатывают их и хранят в собственных базах данных. Операторы, работающие за рабочими станциями, могут соединиться с любым из менеджеров и с помощью графического интерфейса просмотреть данные об управляемой сети, а также выдать менеджеру некоторые директивы по управлению сетью или ее элементами.

Наличие нескольких менеджеров позволяет распределить между ними нагрузку по обработке данных управления, обеспечивая масштабируемость системы. Как правило, используются два типа связей между менеджерами, одноранговая (рис. 23.13) и иерархическая (рис. 23.14).

Рис. 23.13. Одноранговые связи между менеджерами (NE — сетевой элемент)

Рис. 23.14. Иерархические связи между менеджерами (NE — сетевой элемент)

В случае *одноранговых* связей каждый менеджер управляет своей частью сети на основе информации, получаемой от нижележащих агентов. Центральный менеджер отсутствует. Координация работы менеджеров достигается за счет обмена информацией между базами данных менеджеров. Одноранговое построение системы управления сегодня считается неэффективным и устаревшим.

Значительно более гибким является *иерархическое* построение связей между менеджерами. Каждый менеджер нижнего уровня выполняет также функции агента для менеджера верхнего уровня. Такой агент работает уже с укрупненной моделью MIB своей части сети. В такой базе MIB собирается именно та информация, которая нужна менеджеру верхнего уровня для управления сетью в целом.

Модель «менеджер — агент — управляемый объект» лежит в основе таких популярных стандартов управления, как стандарты Интернета на основе протокола SNMP и стандарты управления ISO/OSI на основе протокола CMIP (Common Management Information Protocol — протокол общей управляющей информации).

Более подробную информацию об этом вы можете найти на сайте www.olifer.co.uk в разделе «Системы управления сетью на основе протокола SNMP».

Выводы

С точки зрения пользователей компьютерные сети представляют собой набор служб (сервисов), таких как электронная почта, WWW, интернет-телефония и интернет-телевидение.

Электронная почта — это распределенное приложение, которое построено в архитектуре клиент-сервер и главной функцией которого является предоставление пользователям сети возможности обмениваться электронными сообщениями. Почтовый клиент и почтовый сервер применяют в своей работе специально разработанные для почтовых систем протоколы SMTP, POP3 и IMAP.

Важнейшей сетевой службой является World Wide Web (WWW), или Всемирная паутина; благодаря которой люди получили возможность доступа к огромному объему информации в удобном для них виде и в удобное для них время.

Клиентская часть веб-службы, называемая также браузером, представляет собой приложение, которое устанавливается на компьютере конечного пользователя и одной из важных функций которого является поддержание графического пользовательского интерфейса.

Веб-сервер — это программа, хранящая объекты локально в каталогах компьютера, на котором она запущена, и обеспечивающая доступ к этим объектам по URL-адресам.

Клиент и сервер веб-службы связываются через сеть по протоколу передачи гипертекста HTTP.

IP-телефония — это сервис, который обеспечивает коммутируемые голосовые соединения преимущественно по схеме «один к одному» и который поддерживается сетью, использующей протокол IP в форме общедоступного Интернета или частной IP-сети.

Важнейшим событием в IP-телефонии стало появлением стандартов группы H.323, разработанных ITU-T, и стандартов на основе протокола SIP, разработанных IETF.

Новое поколение IP-телефонии поддерживает широкий спектр услуг, подобный тому, который предоставляют абонентам развитые телефонные сети.

Файловая служба на основе протокола FTP позволяет пользователям удаленных компьютеров обмениваться файлами. FTP-серверы и FTP-клиенты имеются практически в каждой ОС, кроме того, для доступа к FTP-архивам служат FTP-клиенты, встроенные в браузеры.

Системы управления сетью позволяют в автоматическом режиме контролировать сетевой трафик и управлять коммуникационным оборудованием сети. Большинство современных систем управления сетью построены на основе протокола SNMP.

Вопросы и задания

1. Известно, что единственным идентификатором получателя электронной почты, в том числе в схеме с выделенным почтовым сервером, является символьный адрес вида name@domain.com. Каким образом письмо находит путь к почтовому серверу, обслуживающему данного получателя?
2. Заполните таблицу, описывающую свойства почтовых протоколов IMAP, POP3 и SMTP.

Свойство протокола	Протоколы
Используется почтовым клиентом для передачи письма на сервер	
Используется почтовым клиентом для получения письма с сервера	
При получении почты письмо перемещается с сервера на клиент	
При получении почты письмо копируется с сервера на клиент	

3. Браузер находит информацию по адресам специального формата, например такому: http://www.bbc.co.uk/mobile/web/versions.shtml. Поместите в правый столбец таблицы части приведенного адреса, соответствующие названиям в левом столбце.

Путь к объекту	
DNS-имя сервера	
URL-имя	
Тип протокола доступа	

4. Что вы можете сказать о HTTP-сообщении вида HTTP/1.1 200 OK? Варианты ответов:
   а) HTTP-запрос;
   б) HTTP-ответ;
   в) 200 — это код состояния;
   г) 200 — это объем переданной информации;
   д) OK означает, что информация зашифрована открытым ключом;
   е) OK означает, «все в порядке!»
5. Что вы можете сказать о протоколе SIP? Варианты ответов:
   а) протокол веб-службы;
   б) протокол IP-телефонии;
   в) входит в семейство протоколов H.323;
   г) похож на протокол HTTP;
   д) выполняет примерно те же функции, что и протоколы Q.931, RAS и H.245.
6. Что входит в функции привратника? Варианты ответов:
   а) трансляция DNS-имен в телефонные номера;
   б) открытие и закрытие сеанса связи;
   в) регистрация и авторизация абонентов;
   г) маршрутизация вызовов к IP-телефону.
7. Что такое MIB в системе управления сетью? Варианты ответов:
   а) модель управляемого объекта;
   б) база данных управляющей информации;
   в) протокол взаимодействия агента и менеджера системы управления сетью;
   г) набор характеристик объекта, необходимых для его контроля.

ГЛАВА 24 Сетевая безопасность

Обеспечение безопасности названо первой из пяти главных проблем Интернета в программе действий новой международной инициативы построения Интернета будущего (Future Internet Design, FIND). Инициатива FIND направлена на разработку принципов организации того Интернета, который будет служить нам через 15 лет, поэтому участники этой инициативы стараются взглянуть на Интернет свежим взглядом и, возможно, найти новые подходы к его организации.

Сегодня же Интернет представляет собой эффективную, но вместе с тем и непредсказуемую среду, полную разнообразных угроз и опасностей.

Большая группа угроз связана с несовершенством протоколов, в частности протоколов стека TCP/IP. Известно, что эти протоколы разрабатывались в то время, когда проблема обеспечения информационной безопасности еще не стояла на повестке дня. Сообщество пользователей Интернета представляло собой ограниченный круг заинтересованных в эффективной работе Сети специалистов, и уж, конечно, никто не покушался на ее работоспособность. Создаваемые в такой «тепличной» атмосфере протоколы не содержали механизмов, позволяющих противостоять возможным (тогда только теоретически) атакам злоумышленников. Например, хотя в протоколах FTP и telnet и предусмотрена аутентификация, клиент передает пароль серверу по сети в незашифрованном виде, а значит, злоумышленник может перехватить его и получить доступ к FTP-архиву. Сейчас многие из потенциально опасных механизмов, встроенных в протоколы, уже исправлены, и некоторые проблемы, обсуждаемые в этой главе, не являются актуальными, а носят, скорее, исторический и учебный характер.

Многообразие угроз порождает многообразие методов защиты. В этой главе мы будем обсуждать все основные технологии обеспечения информационной безопасности: аутентификацию и авторизацию, шифрование и антивирусные средства, сетевые экраны и прокси-серверы, защищенные каналы и виртуальные частные сети.

Основные понятия информационной безопасности

Определение безопасной системы

Под **информационной безопасностью** понимается состояние защищенности информационной системы, включая собственно информацию и поддерживающую ее инфраструктуру. Информационная система находится в **состоянии защищенности**, если обеспечены ее конфиденциальность, доступность и целостность.

Конфиденциальность (confidentiality) — это гарантия того, что секретные данные будут доступны только тем пользователям, которым этот доступ разрешен; такие пользователи называются легальными, или авторизованными.

Доступность (availability) — это гарантия того, что авторизованные пользователи всегда получат доступ к данным.

Целостность (integrity) — это гарантия сохранности данными правильных значений, которая обеспечивается запретом неавторизованным пользователям каким-либо образом изменять, модифицировать, разрушать или создавать данные.

Требования безопасности могут меняться в зависимости от назначения информационной системы, характера используемых данных и типа возможных угроз. Трудно представить систему, для которой были бы не важны свойства целостности и доступности, но свойство конфиденциальности не всегда является обязательным. Например, если вы публикуете информацию в Интернете на веб-сервере и вашей целью является сделать ее доступной для самого широкого круга людей, конфиденциальность не требуется. Однако требования целостности и доступности остаются актуальными.

Действительно, если вы не предпримете специальных мер по обеспечению целостности системы, злоумышленник может изменить данные на вашем сервере и нанести этим ущерб вашему предприятию. Преступник может, например, внести изменения в помещенный на веб-сервере прайс-лист, что негативно отразится на конкурентоспособности вашего предприятия, или испортить коды свободно распространяемого вашей фирмой программного продукта, что, безусловно, скажется на ее деловой репутации.

Не менее важным в данном примере является и обеспечение доступности данных. Затратив немалые средства на создание и поддержание сервера в Интернете, предприятие вправе рассчитывать на отдачу: увеличение числа клиентов, количества продаж и т. д. Однако существует вероятность того, что злоумышленник предпримет атаку, в результате которой помещенные на сервер данные станут недоступными для тех, кому они предназначались. Примером таких злонамеренных действий может служить «бомбардировка» сервера пакетами, каждый из которых в соответствии с логикой работы соответствующего протокола вызывает тайм-аут сервера, что, в конечном счете, делает его недоступным для всех остальных запросов.

Понятия конфиденциальности, доступности и целостности могут быть определены не только по отношению к информации, но и к другим ресурсам вычислительной сети, таким как внешние устройства или приложения. Так, свойство конфиденциальности по отношению, например, к устройству печати можно интерпретировать так, что доступ к устройству имеют те и только те пользователи, которым этот доступ разрешен, причем они могут выполнять только те операции с устройством, которые для них определены.

Свойство доступности устройства означает его готовность к работе всякий раз, когда в этом возникает необходимость. А свойство целостности может быть определено как свойство неизменности параметров данного устройства.

Легальность использования сетевых устройств важна не только постольку-поскольку она влияет на безопасность данных. Устройства могут предоставлять различные услуги (распечатка текстов, отправка факсов, доступ в Интернет, электронная почта и т. п.), незаконное потребление которых, наносящее материальный ущерб предприятию, также является нарушением безопасности системы.

Угроза, атака, риск

Угроза — любое действие, которое может быть направлено на нарушение информационной безопасности системы.

Атака — реализованная угроза.

Риск — вероятностная оценка величины возможного ущерба, который может понести владелец информационного ресурса в результате успешно проведенной атаки.

Угрозы могут исходить как от легальных пользователей сети, так и от внешних злоумышленников. В последние два года в статистике нарушений безопасности зафиксирован резкий сдвиг от внешних к внутренним угрозам. Примерно 2/3 от общего числа всех наиболее серьезных инцидентов, связанных с безопасностью, составляют нарушения со стороны легальных пользователей сетей: сотрудников и клиентов предприятий, студентов, имеющих доступ к сети учебного заведения и др. Вместе с тем внутренние атаки обычно наносят меньший ущерб, чем внешние.

Угрозы со стороны легальных пользователей делятся на:

- умышленные;
- неумышленные.

К *умышленным* угрозам относятся, например, мониторинг системы с целью получения персональных данных других сотрудников (идентификаторов, паролей) или конфигурационных параметров оборудования. Это может быть также злонамеренное получение доступа к конфиденциальным данным, хранящимся на серверах и рабочих станциях сети «родного» предприятия с целью их похищения, искажения или уничтожения; прямое «вредительство» — вывод из строя сетевого программного обеспечения и оборудования. Кроме того, к умышленным угрозам относится нарушение персоналом правил, регламентирующих работу пользователей в сети предприятия: посещение запрещенных веб-сайтов, вынос за пределы предприятия съемных носителей, небрежное хранение паролей и другие подобные нарушения режима. Однако не меньший материальный ущерб предприятию может быть нанесен в результате *неумышленных* нарушений персонала — ошибок, приводящих к повреждению сетевых устройств, данных, программного обеспечения.

Угрозы внешних злоумышленников, называемых также хакерами, по определению являются умышленными и обычно квалифицируются как преступления. Среди внешних нарушителей безопасности встречаются люди, занимающиеся этой деятельностью профессионально

или просто из хулиганских побуждений. Целью, которой руководствуются внешние злоумышленники, всегда является нанесение вреда предприятию. Это может быть, например, получение конфиденциальных данных, которые могут быть использованы для снятия денег с банковских счетов, или установление контроля над программно-аппаратными средствами сети для последующего их использования в атаках на сети других предприятий.

Как правило, атака предваряется *сбором информации о системе* (mapping), которая помогает не только эффективно спланировать атаку, но и скрыть все следы проникновения в систему. К полезной для хакера информации относятся типы операционных систем и приложений, развернутых в сети, IP-адреса, номера портов клиентских частей приложений, имена и пароли пользователей. Часть информации такого рода может быть получена путем простого общения с персоналом (это называют **социальным инжинирингом**), а часть — с помощью тех или иных программ. Например, определить IP-адреса можно с помощью утилиты ping, задавая в качестве цели адреса из некоторого множества возможных адресов. Если при очередном запуске программы ping пришел ответ, значит, произошло совпадение заданного адреса с адресом узла в атакуемой сети.

Для подготовки и проведения атак могут использоваться либо специально разработанные для этих целей программные средства, либо легальные программы «мирного» назначения. Так, последний пример показывает, как легальная программа ping, которая создавалась в качестве инструмента диагностики сети, может быть применена для подготовки атаки.

При проведении атак злоумышленнику важно не только добиться своей цели, заключающейся в причинении ущерба атакуемому объекту, но и уничтожить все следы своего участия в этом. Одним из основных приемов, используемых злоумышленниками для «заметания следов», является *подмена содержимого пакетов* (spoofing). В частности, для сокрытия места нахождения источника вредительских пакетов (например, при атаке отказа в обслуживании) злоумышленник изменяет значение поля адреса отправителя в заголовках пакетов. Поскольку адрес отправителя генерируется автоматически системным программным обеспечением, злоумышленник вносит изменения в соответствующие программные модули так, чтобы они давали ему возможность отправлять со своего компьютера пакеты с любыми IP-адресами.

Типы и примеры атак

Атаки отказа в обслуживании

Атаки отказа в обслуживании (Denial of Service, DoS) направляются обычно на информационные серверы предприятия, функционирование которых является критически важным условием для работоспособности всего предприятия. Чаще всего объектами DOS-атак становятся основные веб-серверы, файловые и почтовые серверы предприятия, а также корневые серверы системы DNS.

Для проведения DoS-атак злоумышленники часто координируют «работу» нескольких компьютеров (как правило, без ведома пользователей этих компьютеров). Говорят, что в таких случаях имеет место **распределенная атака отказа в обслуживании** (Distributed Denial of Service, DDoS). Злоумышленник, захватив управление над группой удаленных компьютеров, «заставляет» их посылать пакеты в адрес узла-жертвы (рис. 24.1). Получившийся в результате мощный суммарный поток «затопляет» атакуемый компьютер,

вызывая его перегрузку и, в конечном счете, делает его недоступным. Блокировка происходит в результате исчерпания ресурсов либо процессора, либо операционной системы, либо канала связи (полосы пропускания).

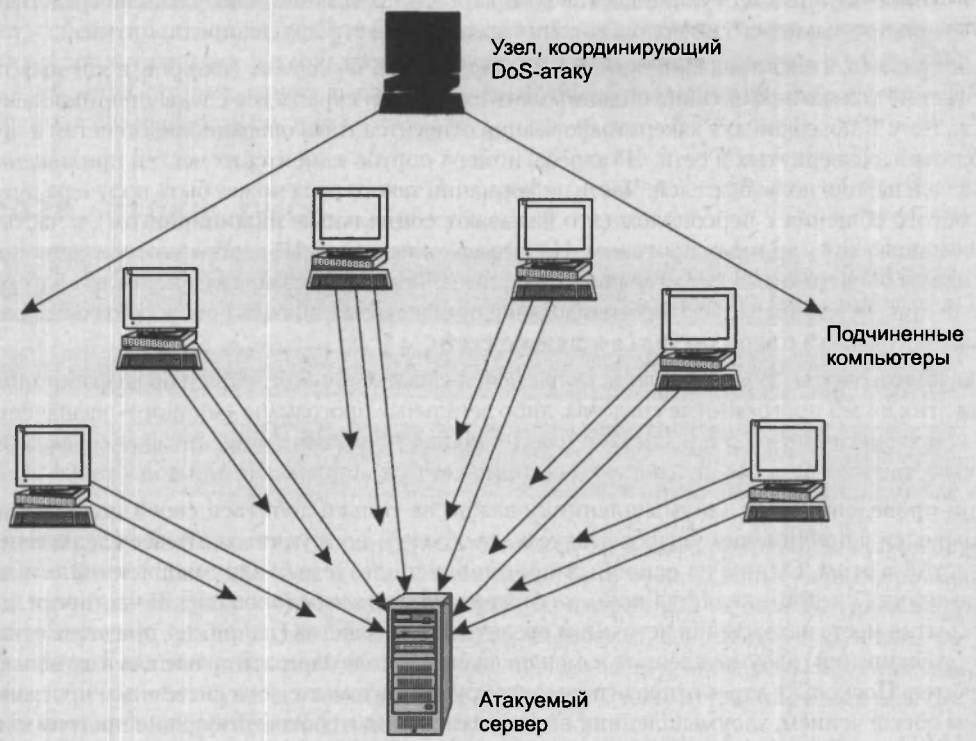


Рис. 24.1. Схема DDoS-атаки

А теперь рассмотрим более конкретный пример проведения DoS-атаки, в которой используются особенности протокола TCP. Как мы уже обсуждали в главе 17, для установления логического соединения по протоколу TCP узлы должны обменяться тремя пакетами (рис. 24.2, *а*): сначала инициатор соединения посылает пакет с флагом *SYN*, на который сервер отвечает пакетом с установленными флагами *ASK* и *SYN*. Завершает процедуру пакет от узла-инициатора с флагом *SYN*.

Для выполнения атаки злоумышленник организует передачу на сервер массированного потока пакетов с флагом *SYN*, каждый из которых инициирует создание нового TCP-соединения (рис. 24.2, *б*). Получив пакет с флагом *SYN*, сервер выделяет для нового соединения необходимые ресурсы и в полном соответствии с протоколом отвечает клиенту пакетом с флагами *ASK* и *SYN*. После этого, установив тайм-аут, он начинает ждать от клиента завершающий пакет с флагом *ASK*, который, увы, так и не приходит. Аналогичным образом создается множество других «недоустановленных» соединений. В результате возникает перегрузка сервера, все его ресурсы идут на поддержание множества соединений, процедуры установления которых остались незавершенными. В таком состоянии сервер уже не способен отвечать на запросы, посылаемые приложениями легальных пользователей, в результате злоумышленник достигает своей цели.

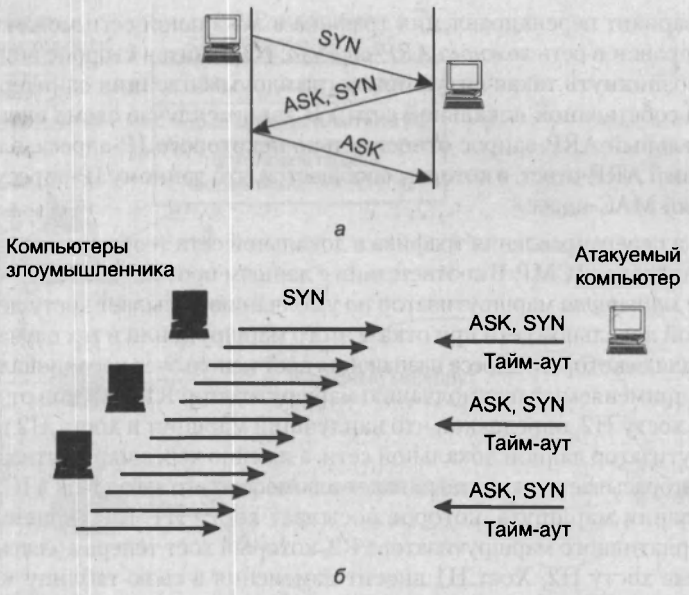


Рис. 24.2. Проведение DoS-атаки, в которой используются особенности протокола TCP: *а* — нормальный порядок установления TCP-соединения; *б* — DDoS-атака за счет создания множества незакрытых TCP-соединений

Подобный подход носит универсальный характер. Например, атака может быть осуществлена путем передачи уязвимому приложению потока запросов, синтаксически правильных, но специально сконструированных, так, чтобы вызвать перегрузку. Так, для некоторых версий веб-сервера Apache губительным оказывается поток запросов, каждый из которых содержит большое количество заголовков HTTP или символов «/».

Перехват и перенаправление трафика

Следующий тип атак имеет целью направить трафик атакуемого компьютера по ложному адресу, в качестве которого может выступать адрес либо злоумышленника, либо третьей стороны. Потоком данных, который пользователь посылает, например, на свой корпоративный сервер или сервер банка, злоумышленник может распорядиться двумя способами. Первый состоит в том, что злоумышленник маскируется под сервера адресата, передавая клиенту ту «картинку» и те сообщения, которые тот ожидает. Так, злоумышленник может имитировать для пользователя-жертвы процедуру логического входа, получая при этом идентификатор и пароль пользователя. Эти данные в дальнейшем могут применяться для несанкционированного доступа к серверу предприятия или банка, которые и являются главной целью атаки. Второй способ заключается в организации транзита трафика. Каждый перехваченный пакет запоминается и/или анализируется на атакующем узле, а после этого переправляется на «настоящий» сервер. Таким образом весь трафик между клиентом и сервером пропускается через компьютер злоумышленника.

Рассмотрим некоторые приемы, используемые сейчас (или в недалеком прошлом) при проведении атак данного типа. Для большинства из них уже разработаны средства противодействия, и приводимые здесь описания атак носят в основном учебный характер.

Простейший вариант перенаправления трафика в локальной сети может быть осуществлен путем отправки в сеть *ложного ARP-ответа*. (Оставим в стороне вопрос, насколько часто может возникнуть такая ситуация, когда злоумышленник заинтересован в перехвате трафика собственной локальной сети.) В данном случае схема очевидна: получив широковещательный ARP-запрос относительно некоторого IP-адреса, злоумышленник посылает ложный ARP-ответ, в котором сообщается, что данному IP-адресу соответствует его собственный MAC-адрес.

Для перехвата и перенаправления трафика в локальной сети теоретически может также использоваться протокол ICMP. В соответствии с данным протоколом *ICMP-сообщение о перенаправлении маршрута* маршрутизатор по умолчанию посылает хосту непосредственно присоединенной локальной сети при отказе этого маршрута или в тех случаях, когда обнаруживает, что для некоторого адреса назначения хост использует нерациональный маршрут. На рис. 24.3, *а* применяемый по умолчанию маршрутизатор R1, получив от хоста H1 пакет, адресованный хосту H2, определяет, что наилучший маршрут к хосту H2 пролегает через другой маршрутизатор данной локальной сети, а именно через маршрутизатор R2. Маршрутизатор R1 отбрасывает полученный пакет и помещает его заголовок в ICMP-сообщение о перенаправлении маршрута, которое посылает хосту H1. В сообщении содержится IP-адрес альтернативного маршрутизатора R2, который хост теперь должен использовать, посылая данные хосту H2. Хост H1 вносит изменения в свою таблицу маршрутизации и с этого момента отправляет пакеты хосту H2 по новому скорректированному маршруту. Для перехвата трафика, направляемого хостом H1 хосту H2, злоумышленник должен сформировать и послать хосту H1 пакет, маскирующийся под ICMP-сообщение о перенаправлении маршрута (рис. 24.3, *б*). В этом сообщении содержится запрос о корректировке таблицы маршрутизации хоста H1, так чтобы во всех пакетах с адресом IP_{H2} адресом следующего маршрутизатора стал адрес IP_{HA}, являющийся адресом хоста-злоумышленника HA. Для того чтобы хост «поверил» этому сообщению, в поле IP-адреса отправителя должен быть помещен адрес маршрутизатора R1, являющегося маршрутизатором по умолчанию. Когда пакеты, передаваемые введенным в заблуждение хостом, начнут поступать на узел злоумышленника, он может либо захватывать и не передавать эти пакеты дальше, имитируя для поддержания диалога приложение, которому эти пакеты предназначались, либо организовать транзитную передачу данных по указанному адресу назначения IP_{H2}. Читая весь трафик между узлами H1 и H2, злоумышленник получает все необходимую информацию для несанкционированного доступа к серверу H2.

Еще одним способом перехвата трафика является использование *ложных DNS-ответов* (рис. 24.4). Задача злоумышленника состоит в получении доступа к корпоративному серверу. Для этого ему нужно завладеть именем и паролем авторизованного пользователя корпоративной сети. Эту информацию он решает получить путем ответвления потока данных, которые корпоративный клиент посылает корпоративному серверу. Злоумышленник знает, что клиент обращается к серверу, указывая его символьное DNS-имя www.example.com. Известно ему также, что перед тем как отослать пакет серверу, программное обеспечение клиентской машины направляет запрос DNS-серверу, чтобы узнать, какой IP-адрес соответствует этому имени.

Цель злоумышленника — опередить ответ DNS-сервера и навязать клиенту свой вариант ответа, в котором вместо IP-адреса корпоративного сервера (в примере 193.25.34.125) злоумышленник указывает IP-адрес атакующего хоста (203.13.1.123). На пути реализации этого плана имеется несколько серьезных препятствий.

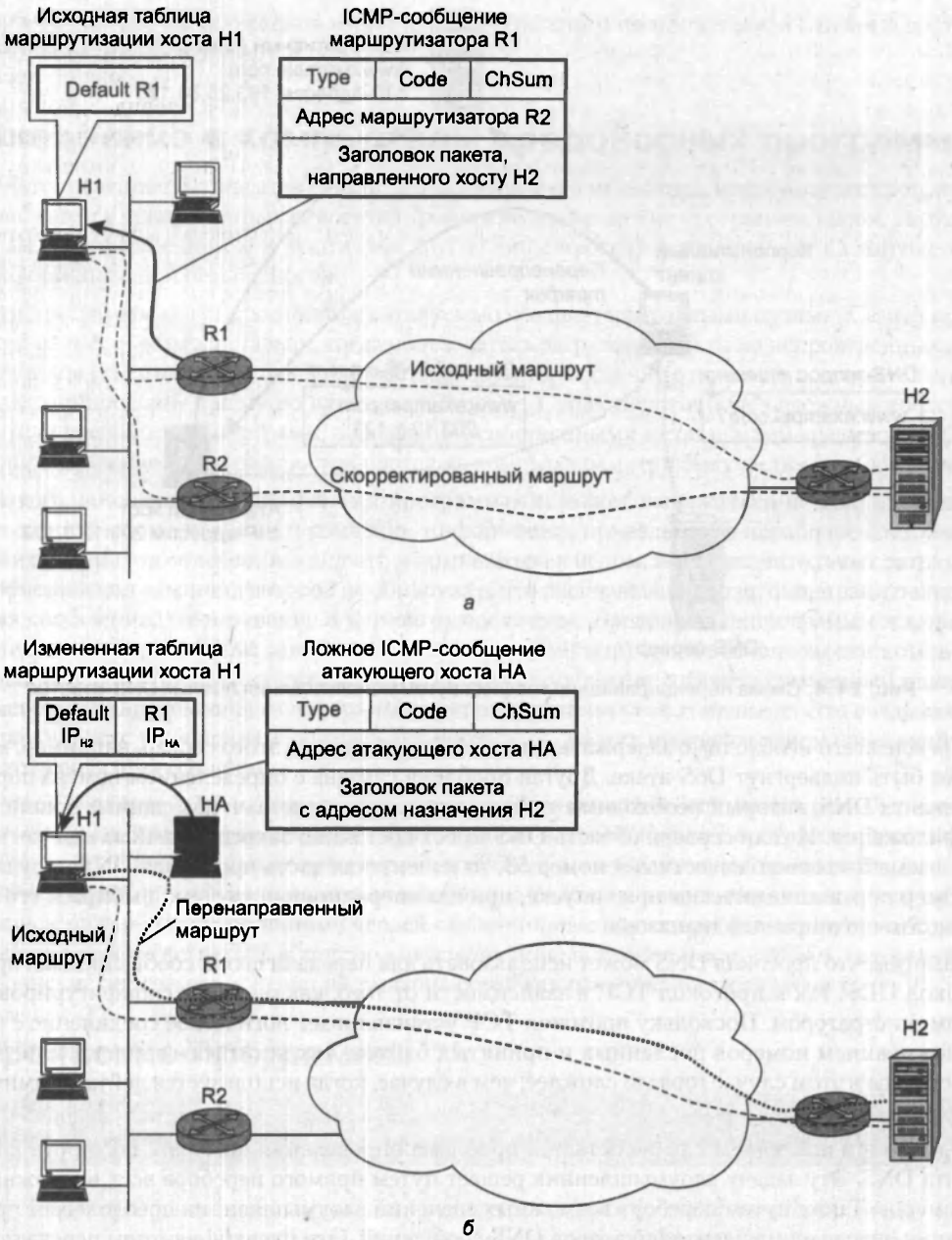


Рис. 24.3. Перенаправление маршрута с помощью протокола ICMP: *а* — сообщение о более рациональном маршруте хосту H2 посылает маршрутизатор R1, применяемый по умолчанию; *б* — сообщение о перенаправлении маршрута на себя направляет атакующий хост HA

Рис. 24.4. Схема перенаправления трафика путем использования ложных DNS-ответов

Прежде всего необходимо задержать ответ DNS-сервера, для этого сервер, например, может быть подвергнут DoS-атаке. Другая проблема связана с определением номера порта клиента DNS, который необходимо указать в заголовке пакета, чтобы данные дошли до приложения. И если серверная часть DNS имеет постоянно закрепленный за ней так называемый «хорошо известный» номер 53, то клиентская часть протокола DNS получает номер порта динамически при запуске, причем операционная система выбирает его из достаточно широкого диапазона.

Заметим, что протокол DNS может использовать для передачи своих сообщений как протокол UDP, так и протокол TCP, в зависимости от того, как он будет сконфигурирован администратором. Поскольку протокол TCP устанавливает логическое соединение с отслеживанием номеров посланных и принятых байтов, «вклиниться» в диалог клиента и сервера в этом случае гораздо сложнее, чем в случае, когда используется дейтаграммный протокол UDP.

Однако и в последнем случае остается проблема определения номера UDP-порта клиента DNS. Эту задачу злоумышленник решает путем прямого перебора всех возможных номеров. Также путем перебора возможных значений злоумышленник преодолевает проблему определения идентификаторов DNS-сообщений. Эти идентификаторы передаются в DNS-сообщениях и служат для того, чтобы клиент системы DNS мог установить соответствие поступающих ответов посланным запросам. Итак, злоумышленник бомбардирует клиентскую машину ложными DNS-ответами, перебирая все возможные значения идентифицирующих полей так, чтобы клиент, в конце концов, принял один из них за истинный DNS-ответ. Как только это происходит, цель злоумышленника можно считать достигнутой — пакеты от клиента направляются на адрес атакующего хоста, злоумышленник

получает в свое распоряжение имя и пароль легального пользователя, а с ними и доступ к корпоративному серверу.

Внедрение в компьютеры вредоносных программ

Многочисленная группа атак связана с внедрением в компьютеры **вредоносных программ** (malware), к числу которых относятся троянские и шпионские программы, черви, вирусы, спам, логические бомбы и некоторые другие типы программ, нацеленные на нарушение информационной безопасности.

Эти программы могут проникать на атакуемые компьютеры разными путями. Самый простой из них — «самодоставка», когда пользователь загружает файлы из непроверенных источников (съемных носителей или веб-сайтов) либо беспечно открывает подозрительный файл, пришедший к нему по электронной почте. Существуют и более сложные представители вредоносных программ, обладающие собственными механизмами «размножения», копии таких программ распространяются по компьютерам сети без участия пользователей.

Ущерб, наносимый вредоносными программами, может выражаться не только в уничтожении, искажении или похищении информации, приведении в нерабочее состояние программного обеспечения, а значит, и компьютера в целом, но и в значительных затратах времени и сил администраторов на обнаружение и распознавание атак, фильтрацию внешних сообщений, тестирование и перезагрузку систем. Вредоносные программы в начале этого десятилетия были одной из основных причин нарушения безопасности компьютерных сетей. Однако как показала статистика, в последние два года суммарный ущерб, нанесенный вредоносными программами предприятиям, резко снизился. Это связывают, в том числе, с улучшением качества антивирусных средств и ужесточением наказаний за такого рода преступления.

Прежде чем перейти к рассмотрению конкретных типов вредоносных программ, заметим, что на практике злоумышленники часто сочетают в одной и той же программе различные типы угроз. Например, некоторые черви способны маскироваться под троянские программы или подобно вирусам заражать исполняемые файлы на локальном диске, а некоторые вирусы наделены способностями червей самокопироваться на другие компьютеры. Кроме того, вы можете встретить и другую классификацию вредоносных программ, где, скажем, троянские программы и черви рассматриваются как разновидности вирусов.

Троянские программы

Троянские программы, или **трояны** (trojan), — это разновидность вредоносных программ, которые наносят ущерб системе, *маскируясь* под какие-либо полезные приложения.

Троянские программы могут применять в качестве прикрытия *знакомые* пользователю приложения, с которыми он работал и раньше, до появления в компьютере «троянского коня». При другом подходе в полном соответствии с древней легендой троянская программа принимает вид *нового* приложения, которое пытается заинтересовать пользователя-жертву какими-то своими якобы полезными функциями.

Однако суть троянской программы и в том и в другом случаях остается вредительской: она может уничтожать или искажать информацию на диске, передавать данные (например, пароли) с «зараженного» компьютера на удаленный компьютер хакера, приводить

в неработоспособное состояние установленное на атакованном компьютере программное обеспечение, участвовать в проведении DoS-атак на другие удаленные компьютеры. Так, одна из известных троянских программ AIDS TROJAN DISK7, разосланная нескольким тысячам исследовательских организаций на дискете, при запуске перемешивала символы в именах всех файлов и заполняла все свободное пространство жесткого диска. После этого программа от имени злоумышленника предлагала помощь в восстановлении диска, требуя взамен вознаграждение для автора этой программы. (Злоумышленники могут также шантажировать пользователя, зашифровывая его данные.) Кстати, описанное компьютерное преступление завершилось поимкой хакера-шантажиста.

Троянские программы могут быть отнесены к самому простому по реализации виду вредоносных программ.

Сетевые черви

Сетевые черви (worm) — это программы, способные к *самостоятельному распространению* своих копий среди узлов в пределах локальной сети, а также по глобальным связям, перемещаясь от одного компьютера к другому без всякого участия в этом процессе пользователей сети.

Поскольку большинство сетевых червей передаются в виде файлов, основным механизмом их распространения являются сетевые службы, основанные на файловом обмене. Так, червь может рассылать свои копии по сети в виде вложений в сообщения электронной почты или путем размещения ссылок на зараженный файл на каком-либо веб-сайте. Однако существуют и другие разновидности червей, которые для своей экспансии используют более сложные приемы, например, связанные с ошибками («дырами») в программном обеспечении.

Главная цель и результат деятельности червя состоит в том, чтобы передать свою копию на максимально возможное число компьютеров. При этом для поиска компьютеров — новых потенциальных жертв — черви задействуют встроенные в них средства. Типичная программа-червь не удаляет и не искажает пользовательские и системные файлы, не перехватывает электронную почту пользователей, не портит содержимое баз данных, а наносит вред атакованным компьютерам путем потребления их ресурсов. Если червь обладает возможностью повторного заражения, то число его копий растет лавинообразно, и вредоносные программы все более и более загружают процессор, захватывая новые области памяти, отбирая пропускную способность сетевых соединений, пока, наконец, программы легальных пользователей не потеряют возможность выполняться.

При создании типичного сетевого червя хакер, прежде всего, определяет перечень сетевых уязвимостей, которые он собирается использовать для проведения атак средствами создаваемого червя. Такими уязвимостями могут быть как известные, но не исправленные на некоторых компьютерах ошибки в программном обеспечении, так и пока неизвестные никому ошибки, которые обнаружил сам хакер. Чем шире перечень уязвимостей и чем более они распространены, тем больше узлов может быть поражено данным червем.

Червь состоит из двух основных функциональных компонентов: атакующего блока и блока поиска целей.

- **Атакующий блок** состоит из нескольких модулей (**векторов атаки**), каждый из которых рассчитан на поражение конкретного типа уязвимости. Этот блок открывает «входную дверь» атакуемого хоста и передает через нее свою копию.

- **Блок поиска целей (локатор)** собирает информацию об узлах сети, а затем на основании этой информации определяет, какие из исследованных узлов обладают теми уязвимостями, для которых хакер имеет средства атаки.

Эти два функциональных блока являются *обязательными* и присутствуют в реализации любой программы-червя. Некоторые черви нагружены их создателями и другими вспомогательными функциями, о которых мы скажем позже.

Упрощенно жизненный цикл червя может быть описан рекурсивной процедурой, состоящей из циклического запуска локатора и атакующего блока на каждом из последующих заражаемых компьютеров (рис. 24.5).

В начале каждого нового цикла червь, базирующийся на захваченном в результате предыдущей атаки компьютере, запускает локатор для поиска и формирования списка узлов-целей, пригодных для проведения каждой из специфических атак, а затем, используя средства атакующего блока, пытается эксплуатировать уязвимости узлов из этого списка. В результате успешной атаки червь копирует все свои программы на «новую территорию» и активирует локатор. После этого начинается новый цикл. На рисунке показано, как червь лавинообразно распространяется по сети. Заражение тысяч компьютеров может занять всего несколько минут. Некоторые виды червей не нападают на уже зараженные и/или подвергающиеся атаке в данный момент узлы. Если же такая проверка не предусмотрена в алгоритме работы червя, то в сети случайным образом могут возникать очаги стихийных DoS-атак.

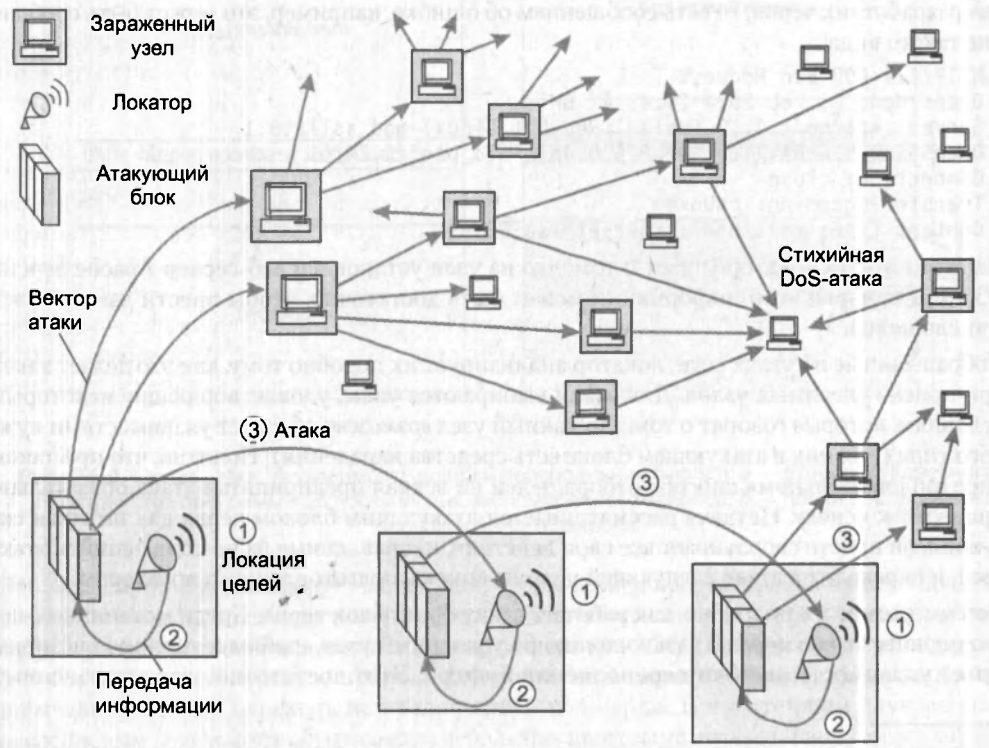


Рис. 24.5. Экспансия червя в сети

Локатор идентифицирует цели по адресам электронной почты, IP-адресам, характеристикам установленных на хостах операционных систем, номерам портов, типам и версиям приложений.

Для сбора информации локатор может предпринимать действия, связанные как с поисками интересующих данных на захваченном им в данный момент хосте, так и путем зондирования сетевого окружения. Простейший способ получить данные локально — прочитать файл, содержащий адресную книгу клиента электронной почты[1]. Помимо почтовых адресов, локатор может найти на узле базирования другие источники информации, такие как таблицы конфигурационных параметров сетевых интерфейсов, ARP-таблицы и таблицы маршрутизации. Зная IP-адреса хоста базирования и шлюзов, локатор достаточно просто может определить IP-адреса других узлов этой сети. Для идентификации узлов локатор может также использовать ICMP-сообщения или запросы ping, указывая в качестве адресов назначения все возможные IP-адреса. Для определения того, какие приложения работают на том или ином хосте, локатор сканирует различные *хорошо известные* номера TCP- и UDP-портов. Определив тип приложения, локатор пытается получить более детальные характеристики этого приложения.

Например, пусть некоторая программа-червь имеет в своем арсенале средства для атаки на некоторые версии веб-сервера Apache. Для поиска потенциальных жертв локатор этого червя зондирует узлы сети, посылая умышленно ошибочные запросы к веб-серверу:

```
GET / HTTP/1.1\r\n\r\n
```

Узел, на котором установлен сервер Apache, отвечает на такой запрос так, как и рассчитывал разработчик червя, то есть сообщением об ошибке, например, это может быть сообщение такого вида:

```
HTTP/1.1 400 Bad Request
Date: Mon, 23 Feb 2004 23:43:42 GMT
Server: Apache/1.3.19 (UNIX) (Red-Hat/Linux) mod_ssl/2.8.1
OpenSSL/0.9.6 DAV/1.0.2 PHP/4.0.4pl1 mod_perl/1.24_01
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1
```

Из этого ответа локатор узнает о том, что на узле установлен веб-сервер Apache версии 1.3.19. Для червя этой информации может быть достаточно, чтобы внести данный узел в число целей.

Собрав данные об узлах сети, локатор анализирует их подобно тому, как это делает хакер при поиске уязвимых узлов. Для атаки выбираются узлы, удовлетворяющие некоторым условиям, которые говорят о том, что данный узел *возможно* обладает уязвимостями нужного типа (для них в атакующем блоке есть средства нападения). Понятно, что при таком «предположительном» способе отбора целей не всякая предпринятая атака обязательно приводит к успеху. Неудача рассматривается атакующим блоком червя как штатная ситуация, он просто сворачивает все свои действия, направленные на не поддавшийся атаке узел, и переходит к атаке следующей цели из списка, подготовленного локатором.

Рассмотрим более подробно, как работает атакующий блок червя. Среди механизмов, позволяющих червю передать свою копию на удаленный узел, наиболее длинную историю имеет уязвимость **ошибки переполнения буфера**. Этот достаточно распространенный

---

[1] Для коллекционирования почтовых адресов локатор может прибегать и к более интеллектуальным методам, которые используют в своей работе спамеры (о спаме см. далее).

вид уязвимости связан с неправильной работой некоторых программ, когда у них переполняется буфер.

При трансляции программ, написанных на многих языках программирования, в исполняемом (объектном) модуле в сегменте локальных переменных отводится место для буферов, в которые будут загружаться данные при выполнении процедур ввода. Например, в программе веб-сервера должен быть предусмотрен буфер для размещения запросов, поступающих от клиентов. Причем размер буфера должен быть равен максимально допустимой для данного протокола длине запроса. В том же сегменте локальных переменных транслятор размещает команду возврата из процедуры, которой будет передано управление при завершении процедуры (рис. 24.6, *а*).

Для правильной работы программы очень важно, чтобы вводимые данные (в нашем примере — запрос клиента) всегда укладывались в границы отведенного для них буфера. В противном случае эти данные записываются поверх команды возврата из процедуры. А это, в свою очередь, означает, что процедура не сможет завершиться корректно: при передаче управления на адрес команды возврата процессор будет интерпретировать в качестве команды то значение из запроса, которое записано поверх команды возврата. Если такого рода переполнение возникло в результате случайной ошибки, то маловероятно, что значение, записанное поверх команды возврата, окажется каким-либо осмысленным кодом. Иное дело, если это переполнение было специально инициировано злоумышленником.

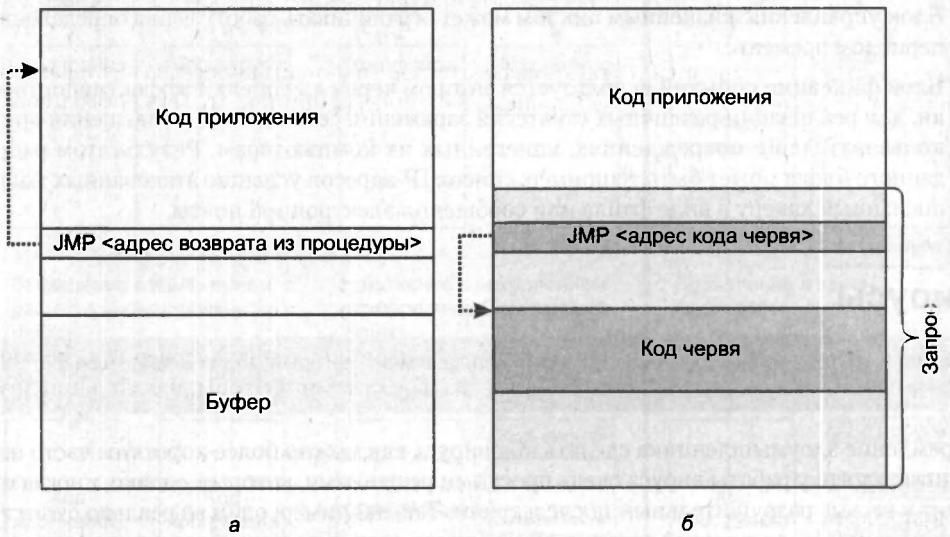


Рис. 24.6. Схема атаки на уязвимость ошибки переполнения буфера: *а* — структура адресного пространства программы до поступления злонамеренного запроса; *б* — после поступления злонамеренного запроса

Злоумышленник конструирует запрос так, чтобы сервер прореагировал на него предсказуемым и желательным для хакера образом. Для этого хакер посылает нестандартный запрос, размер которого превышает размер буфера (рис. 24.6, *б*). При этом среди данных запроса в том месте, которое приходится как раз на команду возврата, злоумышленник помещает команду перехода на вредоносный код червя. В простейшем случае таким вредоносным кодом может быть совсем небольшая программа, переданная в том же запросе.

Итак, атакующий блок червя посылает некорректный запрос уязвимому серверу, его буфер переполняется, код команды возврата из процедуры замещается кодом команды передачи управления вредоносной программе, которая выполняет копирование всех оставшихся программных модулей червя на вновь освоенную территорию.

Хотя рассмотренный подход применим к самым различным приложениям, для каждого типа приложений хакер должен сформировать специальный атакующий запрос, в котором *смещение кода команды передачи управления вредоносной программе точно соответствовало бы местоположению команды возврата в процедуру атакуемого приложения*. Именно поэтому для червя при проведении такого вида атак так важно получить информацию о типе и версиях программного обеспечения, установленного на узлах сети.

Помимо локатора и атакующего блока червь может включать некоторые дополнительные функциональные компоненты.

- **Блок удаленного управления и коммуникаций** служит для передачи сетевым червям команд от их создателя, а также для взаимодействия червей между собой. Такая возможность позволяет хакеру координировать работу червей для организации распределенных атак отказа в обслуживании. Сетевые черви могут быть также использованы для организации параллельных вычислений при решении таких требующих большого объема вычислений задач, как, например, подбор секретного ключа шифрования или пароля.
- **Блок управления жизненным циклом** может ограничивать работу червя определенным периодом времени.
- **Блок фиксации событий** используется автором червя для оценки эффективности атаки, для реализации различных стратегий заражения сети или для оповещения других пользователей о повреждениях, нанесенных их компьютерам. Результатом работы данного блока может быть, например, список IP-адресов успешно атакованных машин, посланный хакеру в виде файла или сообщения электронной почты.

Вирусы

Вирус (virus) — это вредоносный программный фрагмент, который может внедряться в другие файлы.

Стремление злоумышленника сделать код вируса как можно более коротким часто ограничивает логику работы вируса очень простыми решениями, которые, однако, иногда приводят к весьма разрушительным последствиям. Так, например, один из реально существовавших вирусов, состоящий всего из 15 (!) байтов, записывал свою копию поверх других файлов в начало каждого сектора диска, в результате система очень быстро терпела крах. Некоторым утешением в таком и подобных ему случаях является то, что одновременно с крахом компьютера прекращает свое существование и вирус.

Вирус может внедрять свои фрагменты в разные типы файлов, в том числе в файлы исполняемых программ (рис. 24.7). При этом возможны самые разные варианты: замещение кода, когда размер инфицированного файла не меняется, вставка вирусного кода целиком в начало или конец исходной программы, замена фрагментов программного кода фраг-

ментами вируса с перестановкой замещенных фрагментов и без перестановки и т. д., и т. п. Более того, код вируса может быть зашифрован, чтобы затруднить его обнаружение антивирусными программами.

В отличие от червей вирусы (так же как и троянские программы) не содержат в себе встроенного механизма активного распространения по сети, они способны размножаться своими силами только в *пределах одного компьютера*. Как правило, передача копии вируса на другой компьютер происходит *с участием пользователя*. Например, пользователь может записать свой файл, зараженный вирусом, на сетевой файловый сервер, откуда тот может быть скопирован всеми пользователями, имеющими доступ к данному серверу. Пользователь может также передать другому пользователю съемный носитель с зараженным файлом или послать такой файл по электронной почте. То есть именно пользователь является главным звеном в цепочке распространения вируса за пределы своего компьютера. Тяжесть последствий вирусного заражения зависит от того, какие вредоносные действия были запрограммированы в вирусе злоумышленником. Это могут быть мелкие, но раздражающие неудобства (замедление работы компьютера, уменьшение размеров доступной памяти, трата рабочего времени на переустановку приложений) или серьезные нарушения безопасности, такие как утечка конфиденциальных данных, разрушение системного программного обеспечения, частичная или полная потеря работоспособности компьютерной сети.

Рис. 24.7. Различные варианты расположения кода вируса в зараженных файлах

Шпионские программы

Шпионские программы (spyware) — это такой тип вредоносных программ, которые тайно (как правило, удаленно) устанавливаются злоумышленниками на компьютеры ничего не подозревающих пользователей, чтобы отслеживать и фиксировать все их действия.

В число таких действий может входить введение имени и пароля во время логического входа в систему, посещение тех или иных веб-сайтов, обмен информацией с внешними и внутренними пользователями сети и пр., и пр. Собранная информация пересылается злоумышленнику, который применяет ее в преступных целях.

Заметим, что в качестве шпионских программ могут использоваться не только созданные специально для этих целей вредоносные программы, но и программы легального назначения. Так, опасным средством шпионажа могут стать легальные системы мониторинга сети[1], такие, например, как популярные сетевые мониторы Wireshark или Microsoft Network Monitor. Исходное назначение этих программ состоит в том, чтобы дать администратору сети возможность следить за сетевым трафиком, в частности захватывать пакеты, используя механизм фильтрации, просматривать их содержимое, собирать статистику по загрузке устройств. В руках же злоумышленника такая программа превращается в мощный инструмент «взлома» сети, который позволяет перехватывать пакеты с паролями и другой секретной информацией. Они также позволяют путем сканирования TCP- и UDP-портов определять типы приложений, работающих в сети, что является очень важной информацией для подготовки атаки.

ПРИМЕЧАНИЕ

Практически все сетевые мониторы построены в архитектуре клиент-сервер. Клиенты, обычно называемые агентами, захватывают и, если необходимо, фильтруют трафик, а затем передают его серверной части монитора для дальнейшей обработки. Серверная часть монитора может работать как в локальной сети, так и на удаленном компьютере, однако клиентские части всегда устанавливаются на компьютерах в тех сегментах сети, в которых протекает интересующий администратора (или злоумышленника) трафик. Необходимым условием для работы агентов монитора является установка сетевого адаптера компьютера, на котором запущен этот агент, в неразборчивый режим приема (см. раздел «MAC-адреса» в главе 12). Поэтому одним из способов, пресекающих несанкционированный захват и анализ сетевого трафика, является отслеживание всех интерфейсов сети, работающих в неразборчивом режиме приема.

Спам

Спам[2] — это атака, выполненная путем злоупотребления возможностями электронной почты.

Учитывая ту важную роль, которую играет электронная почта в работе современных предприятий и организаций, можно понять, почему спам, дезорганизующий работу этой службы, стал рассматриваться в последние годы как одна из существенных угроз безопасности.

---

[1] Программные системы, предназначенные для анализа сетевого трафика, называют также снифферами (sniffers от английского sniff – нюхать).

[2] Спам получил свое название по имени реально существующих консервов Spam, которые стали темой одного из эпизодов популярного английского сериала. В этом эпизоде посетители кафе страдают оттого, что им постоянно навязывают блюда, в которых присутствуют эти консервы.

Спам отнимает время и ресурсы на просмотр и удаление бесполезных сообщений, при этом ошибочно могут быть удалены письма с критически важной информацией, особенно велика вероятность этого при автоматической фильтрации писем. Посторонняя почта, которая нередко составляет 70 % получаемых сообщений, не только снижает эффективность работы предприятия, но и зачастую служит средством внедрения вредоносных программ. Кроме того, спам часто является элементом различных мошеннических схем, жертвами которых могут стать как отдельные сотрудники, так и предприятие в целом.

Спамеры, то есть лица, рассылающие спам, используют для своих целей разнообразные и иногда весьма сложные методы и средства. Так, например, для пополнения баз данных адресов ими может выполняться автоматическое сканирование страниц Интернета, а для организации массовой рассылки они могут прибегать к распределенным атакам, когда зомбированные с помощью червей компьютеры бомбардируют спамом огромное число пользователей сети.

Методы обеспечения информационной безопасности

Обеспечение информационной безопасности — это деятельность, направленная на достижение состояния защищенности (целостности, конфиденциальности и доступности) информационной среды, а также на прогнозирование, предотвращение и смягчение последствий любых случайных или злонамеренных воздействий, результатом которых может явиться нанесение ущерба самой информации, ее владельцам или поддерживающей инфраструктуре.

Классификация методов защиты

Сегодня существует большой арсенал методов обеспечения информационной безопасности, к которым мы, прежде всего, отнесем *технические средства защиты*, такие как системы шифрования, аутентификации, авторизации, аудита, антивирусной защиты, межсетевые экраны и др. Именно им в основном посвящена данная глава этого учебника.

Однако помимо технических средств, не меньшее, а иногда и большее влияние на безопасность системы оказывают средства, построенные на качественно другой основе. К таким «не техническим» мерам защиты относятся соответствующие *сфера законодательства*, *морально-этические нормы*, *просветительная работа и административные меры*. Например, именно ужесточением наказаний за преступления в области нарушения информационной безопасности эксперты объясняют резкое снижение за последние два года количества вирусных атак. Примером эффективных административных мер может служить запрет сотрудникам пользоваться в пределах предприятия собственными ноутбуками; такой запрет сокращает случаи утечки конфиденциальной информации и заражения корпоративных данных новыми вирусами.

Важную роль играют также *физические средства защиты*, к которым относят замки, камеры наблюдения, охранные системы. Данные, записанные на съемный носитель, помещенный в сейф в хорошо охраняемом помещении, очевидно, более защищены, чем данные, хранящиеся на диске работающего в сети компьютера, защищенного самым совершенным сетевым экраном[1].

---

[1] См. далее раздел «Сетевые экраны».

Универсальным средством противодействия атакам, имеющим целью нарушение целостности данных, а в некоторых случаях и их доступности, является резервное копирование. **Резервное копирование** — это набор автоматизированных процедур создания и поддержания копий данных, которые могут быть использованы для восстановления исходных данных в случае их потери или искажения. Резервные копии записывают на сменные носители большой емкости, например магнитные ленты, которые для повышения отказоустойчивости размещают в местах, территориально разнесенных с местонахождением исходных данных. Понятно, что при этом возрастает вероятность их потери или кражи. Чтобы смягчить возможные последствия этих угроз, копируемые данные записываются на сменные носители в зашифрованном виде.

Для эффективного поддержания информационной безопасности необходим *системный подход*. Это означает, что различные средства защиты (технические, юридические, административные, физические и т. д.) должны применяться совместно и под централизованным управлением.

Политика безопасности

Организация служб безопасности сети требует тщательной проработки **политики информационной безопасности**, которая включает несколько базовых принципов.

Одним из таких принципов является предоставление каждому сотруднику предприятия того *минимального уровня привилегий* на доступ к данным, который необходим ему для выполнения его должностных обязанностей.

Следующий принцип — использование *многоуровневого подхода* к обеспечению безопасности. Система защиты с многократным резервированием средств безопасности увеличивает вероятность сохранности данных. Так, например, физические средства защиты (закрытые помещения, блокировочные ключи), ограничивающие непосредственный контакт пользователя только приписанным ему компьютером, дополняют и усиливают эффективность централизованной системы авторизации пользователей.

Принцип *единого контрольно-пропускного пункта* заключается в том, что весь входящий во внутреннюю сеть и выходящий во внешнюю сеть трафик проходит через единственный узел сети, например через сетевой экран. Только это позволяет в достаточной степени контролировать трафик. В противном случае, когда в сети имеется множество пользовательских станций, имеющих независимый выход во внешнюю сеть, очень трудно скоординировать правила, ограничивающие права пользователей внутренней сети на доступ к серверам внешней сети и обратно — права внешних клиентов на доступ к ресурсам внутренней сети.

Используя многоуровневую систему защиты, важно обеспечивать *баланс надежности защиты всех уровней*. Если в сети все сообщения шифруются, но ключи легкодоступны, то эффект от шифрования нулевой. Если внешний трафик сети, подключенной к Интернету, проходит через мощный сетевой экран, но пользователи имеют возможность связываться с узлами Интернета по коммутируемым линиям через локально установленные модемы, то деньги (как правило, немалые), потраченные на сетевой экран, можно считать выброшенными на ветер.

Следующим универсальным принципом является использование только таких средств, которые при отказе переходят в состояние *максимальной защиты*. Это касается самых различных средств безопасности. Если в сети имеется устройство, которое анализирует весь входной трафик и отбрасывает кадры с определенным, заранее заданным обратным адресом, то при отказе оно должно полностью блокировать вход в сеть. Неприемлемым

следовало бы признать устройство, которое при отказе просто отключается, начиная пропускать в сеть весь внешний трафик.

Следующим является принцип *баланса возможного ущерба от реализации угрозы и затрат на ее предотвращение*. Ни одна система безопасности не гарантирует защиту данных на уровне 100 %, поскольку является результатом компромисса между возможными рисками и возможными затратами. Определяя политику безопасности, администратор должен взвесить величину ущерба, которую может понести предприятие в результате нарушения защиты данных, и соотнести ее с величиной затрат, требуемых на обеспечение безопасности этих данных. Так, в некоторых случаях можно отказаться от дорогостоящего межсетевого экрана в пользу стандартных средств фильтрации обычного маршрутизатора, в других же приходится идти на беспрецедентные затраты. Главное, чтобы принятое решение было обосновано экономически.

НЕМНОГО СТАТИСТИКИ

По данным отчета[1] о состоянии информационной безопасности на предприятиях и компаниях Великобритании в 2008 году подавляющее большинство предприятий использует средства защиты, а именно:

- 99 % регулярно выполняют резервное копирование своих наиболее важных данных;
- 98 % имеют средства обнаружения шпионских программ;
- 97 % фильтруют трафик электронной почты на наличие спама;
- 97 % используют сетевые экраны для защиты своих веб-сайтов;
- 95 % сканируют входящие сообщения электронной почты на предмет содержания в них вирусов;
- 94 % шифруют трафик своих беспроводных сетей.

Шифрование

Шифрование — это средство обеспечения конфиденциальности данных, хранящихся в памяти компьютера или передаваемых по проводной или беспроводной сети.

Шифрование является краеугольным камнем всех служб информационной безопасности, будь то система аутентификации или авторизации, защищенный канал или средства безопасного хранения данных.

Любая процедура шифрования, превращающая информацию из обычного «понятного» вида в «нечитабельный» зашифрованный, естественно должна быть дополнена процедурой **дешифрирования**, которая, будучи примененной к зашифрованному тексту[2], снова приводит его в понятный вид.

Пара процедур — шифрование и дешифрирование — называется **криптосистемой**. Обычно криптосистема предусматривает наличие специального параметра — **секретного ключа**. Криптосистема считается *раскрытой*, если найдена процедура, позволяющая подобрать ключ за реальное время. Сложность алгоритма раскрытия является одной из важных характеристик криптосистемы и называется **криптостойкостью**.

---

[1] См. отчет «О нарушениях информационной безопасности 2008» («The Information Security Breaches Survey 2008»), представленный компанией PricewaterhouseCoopers по поручению Министерства предпринимательства, промышленности и управленческих реформ Великобритании.

[2] Информацию, над которой выполняются функции шифрования и дешифрирования, будем условно называть *текстом*, учитывая, что это может быть также числовой массив или графические данные.

В криптографии принято **правило Керкхоффа**, заключающееся в том, что *стойкость шифра должна определяться только секретностью ключа*. Так, все стандартные алгоритмы шифрования (например, AES, DES, PGP) широко известны[1], их детальное описание содержится в легкодоступных документах, но от этого их эффективность не снижается. Система остается защищенной, если злоумышленнику известно все об алгоритме шифрования, но он не знает секретный ключ.

Существует два класса криптосистем — *симметричные* и *асимметричные*. В симметричных схемах шифрования (классическая криптография) секретный ключ шифрования совпадает с секретным ключом дешифрирования. В асимметричных схемах шифрования (криптография с открытым ключом) открытый ключ шифрования не совпадает с секретным ключом дешифрирования.

Симметричные алгоритмы шифрования

На рис. 24.8 приведена классическая модель **симметричной криптосистемы**, теоретические основы которой впервые были изложены в 1949 году в работе Клода Шеннона. В данной модели три участника: отправитель, получатель и злоумышленник. Задача отправителя заключается в том, чтобы по открытому каналу передать некоторое сообщение в защищенном виде. Для этого он зашифровывает открытый текст X ключом k и передает шифрованный текст Y. Задача получателя заключается в том, чтобы расшифровать Y и прочитать сообщение X. Предполагается, что отправитель имеет свой источник ключа. Сгенерированный ключ заранее по надежному каналу передается получателю. Задача злоумышленника заключается в перехвате и чтении передаваемых сообщений, а также в имитации ложных сообщений.

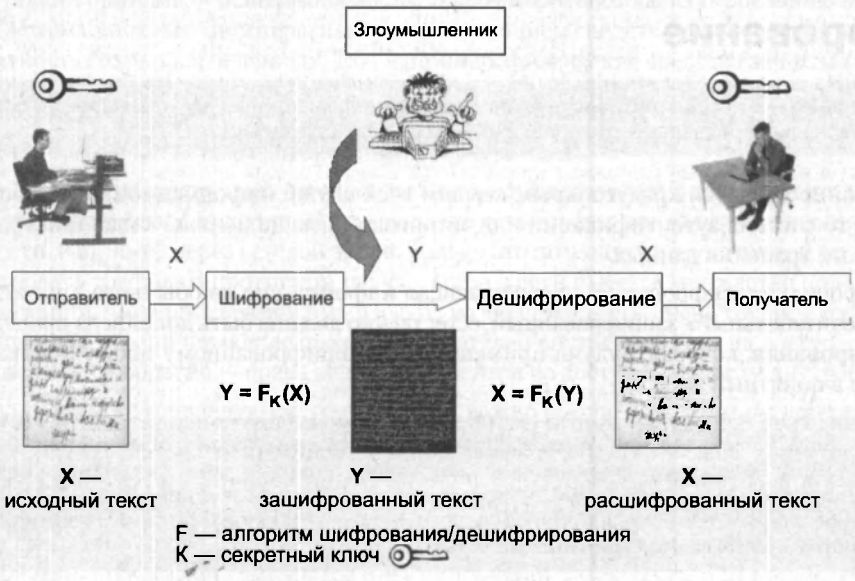


Рис. 24.8. Модель симметричного шифрования

[1] Следует отметить, однако, что существует немало фирменных алгоритмов, описание которых не публикуется.

Модель является универсальной — если зашифрованные данные хранятся в компьютере и никуда не передаются, отправитель и получатель совмещаются в одном лице, а в роли злоумышленника выступает некто, имеющий доступ к компьютеру в ваше отсутствие.

Алгоритм DES

Наиболее популярным стандартным симметричным алгоритмом шифрования данных является **DES** (Data Encryption Standard). Алгоритм разработан фирмой IBM и в 1976 году был рекомендован Национальным бюро стандартов к использованию в открытых секторах экономики. Суть этого алгоритма заключается в следующем (рис. 24.9).

Рис. 24.9. Схема шифрования по алгоритму DES

Данные шифруются *поблочно*. Перед шифрованием любая форма представления данных преобразуется в числовую. Числа получают путем применения любой открытой процедуры преобразования блока текста в число. Например, ими могли бы быть значения двоичных чисел, полученных слиянием кодов ASCII последовательных символов соответствующего блока текста. На вход шифрующей функции поступает блок данных размером 64 бита, он делится пополам на левую (L) и правую (R) части. На первом этапе на место левой части результирующего блока помещается правая часть исходного блока. Правая часть результирующего блока вычисляется как сумма по модулю 2 (операция XOR) левой и правой частей исходного блока. Затем на основе случайной двоичной последовательности по определенной схеме в полученном результате выполняются побитные замены и перестановки. Используемая двоичная последовательность, представляющая собой ключ данного алгоритма, имеет длину 64 бита, из которых 56 действительно случайны, а 8 предназначены для контроля ключа.

Вот уже более трех десятков лет алгоритм DES испытывается на стойкость. И хотя существуют примеры успешных попыток «взлома» данного алгоритма, в целом можно считать, что он выдержал испытания. Алгоритм DES широко используется в различных технологиях и продуктах, связанных с безопасностью информационных систем. Для того чтобы повысить криптостойкость алгоритма DES, иногда применяют его усиленный вариант, называемый «тройным алгоритмом DES», который включает троекратное шифрование с использованием двух разных ключей. При этом можно считать, что длина ключа увеличивается с 56 до 112 бит, а значит, криптостойкость алгоритма существенно повышается. Но за это приходится платить производительностью — тройной алгоритм DES требует в три раза больше времени на реализацию, чем «обычный».

В 2001 году Национальное бюро стандартов США приняло новый стандарт симметричного шифрования, который получил название **AES** (Advanced Encryption Standard). Стандарт AES был разработан в результате проведения конкурса на разработку симметричного алгоритма

шифрования, обладающего лучшим, чем у DES, сочетанием показателей безопасности и скорости работы. Победителем был признан алгоритм Rijndael, который и был положен в основу AES. В результате AES обеспечивает лучшую защиту, так как использует 128-битные ключи (а также может работать со 192- и 256-битными ключами) и имеет более высокую скорость работы, кодируя за один цикл 128-битный блок в отличие от 64-битного блока DES. В настоящее время AES является наиболее распространенным симметричным алгоритмом шифрования.

В симметричных алгоритмах главную проблему представляют ключи. Во-первых, криптостойкость многих симметричных алгоритмов зависит от качества ключа, это предъявляет повышенные требования к службе генерации ключей. Во-вторых, принципиальной является надежность канала передачи ключа второму участнику секретных переговоров. Проблема с ключами возникает даже в системе с двумя абонентами, а в системе с n абонентами, желающими обмениваться секретными данными по принципу «каждый с каждым», потребуется $n \times (n-1)/2$ ключей, которые должны быть сгенерированы и распределены надежным образом. То есть количество ключей пропорционально квадрату количества абонентов, что при большом числе абонентов делает задачу чрезвычайно сложной. Несимметричные алгоритмы, основанные на использовании открытых ключей, снимают эту проблему.

Несимметричные алгоритмы шифрования

В середине 70-х двое ученых — Винфилд Диффи и Мартин Хеллман — описали принципиально другой подход к шифрованию.

Особенность **шифрования с открытым ключом** состоит в том, что одновременно генерируется уникальная *пара ключей*, таких что текст, зашифрованный одним ключом, может быть расшифрован только с использованием второго ключа, и наоборот.

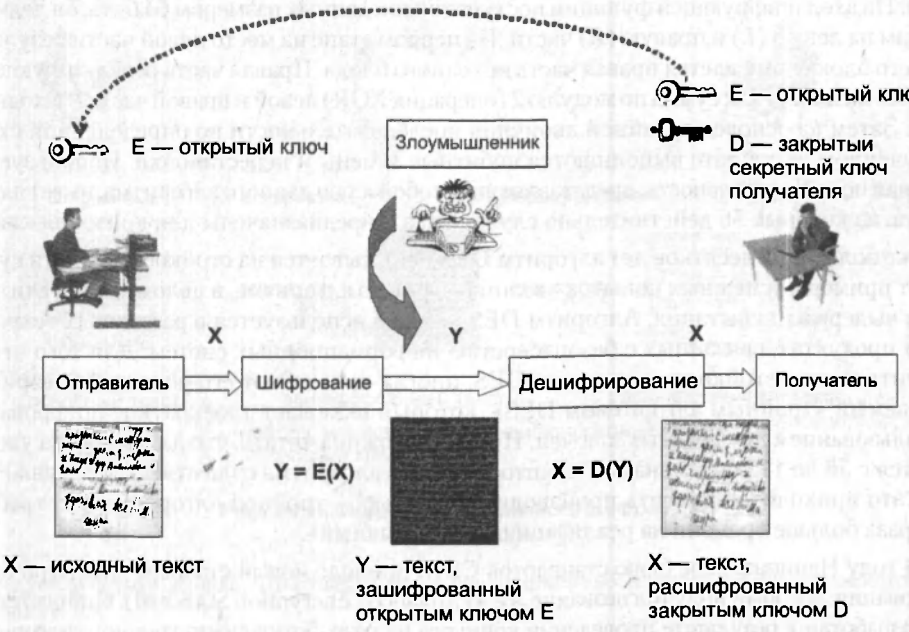


Рис. 24.10. Модель криптосхемы с открытым ключом

В модели криптосхемы с открытым ключом также три участника: отправитель, получатель и злоумышленник (рис. 24.10). Задача отправителя заключается в том, чтобы по открытому каналу связи передать некоторое сообщение в защищенном виде. Получатель генерирует на своей стороне два ключа: открытый *E* и закрытый *D*. Закрытый ключ *D* (часто называемый также личным ключом) абонент должен сохранять в защищенном месте, а открытый ключ *E* он может передать всем, с кем хочет поддерживать защищенные отношения. Для шифрования текста служит открытый ключ, но расшифровать этот текст можно только с помощью закрытого ключа. Поэтому открытый ключ передается отправителю в незащищенном виде. Отправитель, используя открытый ключ получателя, шифрует сообщение *X* и передает его получателю. Получатель расшифровывает сообщение своим закрытым ключом *D*. Очевидно, что числа, одно из которых служит для шифрования текста, а другое — для дешифрирования, не могут быть независимыми друг от друга, а значит, есть теоретическая возможность вычисления закрытого ключа по открытому. Однако это связано с огромным объемом вычислений, которые требуют соответственно огромного времени. Поясним принципиальную связь между закрытым и открытым ключами следующей аналогией.

ПРИМЕР-АНАЛОГИЯ

Пусть руководитель предприятия (на рис. 24.11 это пользователь 1) решает вести секретную переписку со своими сотрудниками. Рассмотрим вариант, когда требуется обеспечить конфиденциальность потока сообщений только в одну сторону — от сотрудников к руководителю. Для этого руководитель решает использовать какой-либо малоизвестный язык, например санскрит. С этой целью он обзаводится единственной копией санскритско-русского словаря, который оставляет себе, и большим количеством широкодоступных русско-санскритских словарей, которые раздает всем своим сотрудникам.

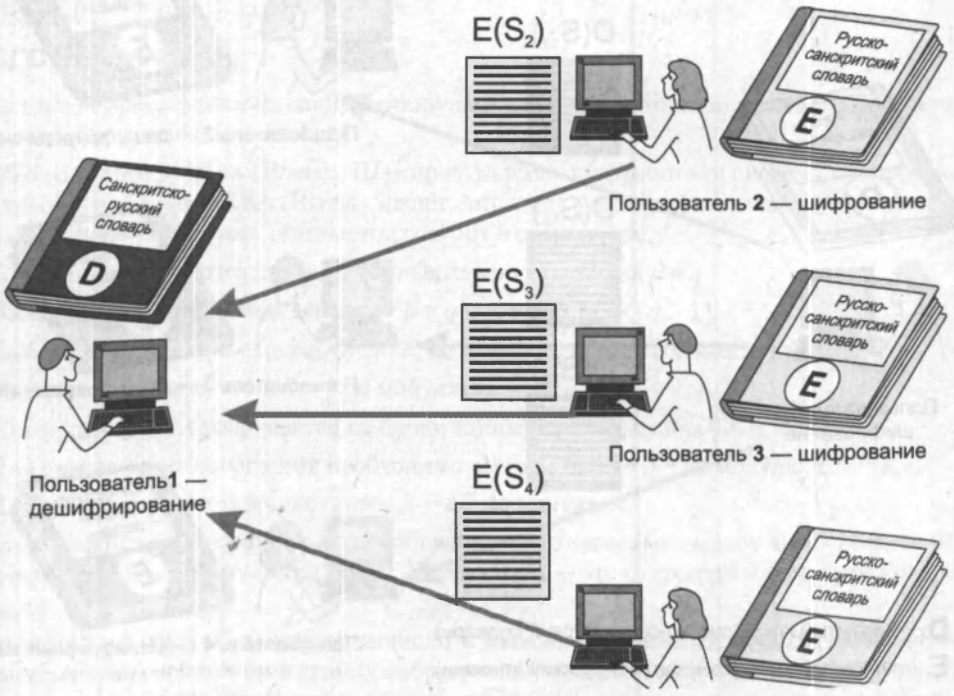


Рис. 24.11. Использование шифрования с открытым ключом для обеспечения конфиденциальности

Когда у сотрудников возникает необходимость написать секретное сообщение руководителю, они, пользуясь словарем, пишут сообщения на санскрите. Руководитель переводит сообщения на русский язык, пользуясь доступным *только ему* санскритско-русским словарем. Очевидно, что здесь роль открытого ключа E и закрытого ключа D руководителя играют русско-санскритский и санскритско-русский словари соответственно. Могут ли пользователи 2, 3 и 4 прочитать чужие сообщения S_2, S_3, S_4, которые посылает каждый из них руководителю? Вообще-то нет, так как для этого им нужен санскритско-русский словарь, обладателем которого является только пользователь 1. Так обеспечивается конфиденциальность потока сообщений в направлении руководителя.

Заметим, что у сотрудников имеется теоретическая возможность для разгадывания сообщений друг друга, так как, затратив массу времени, можно прямым перебором составить санскритско-русский словарь по русско-санскритскому. Такая очень трудоемкая процедура, требующая больших затрат времени, отдаленно напоминает восстановление закрытого ключа по открытому.

На рис. 24.12 показана другая схема использования открытого и закрытого ключей, целью которой является подтверждение авторства (аутентификация) посылаемого сообщения. Пусть задача подтверждения авторства ставится только в отношении посланий руководителя своим сотрудникам. В этом случае роль закрытого (*D*) и открытого (*E*) ключей руководителя играют русско-санскритский и санскритско-русский словари соответственно, причем наши предположения о доступности этих словарей меняются на противоположные. Итак, руководитель пишет письма своим сотрудникам на санскрите (то есть шифрует их закрытым ключом *D*). Сотрудник, получивший послание, пытается перевести зашифрованную часть письма, пользуясь санскритско-русским словарем (открытым ключом *E*). Если ему это удается, то это доказывает, что текст был зашифрован закрытым ключом, парным открытому ключу *E* руководителя. А владельцем этого парного ключа может быть только руководитель, значит, именно он является автором этого сообщения.

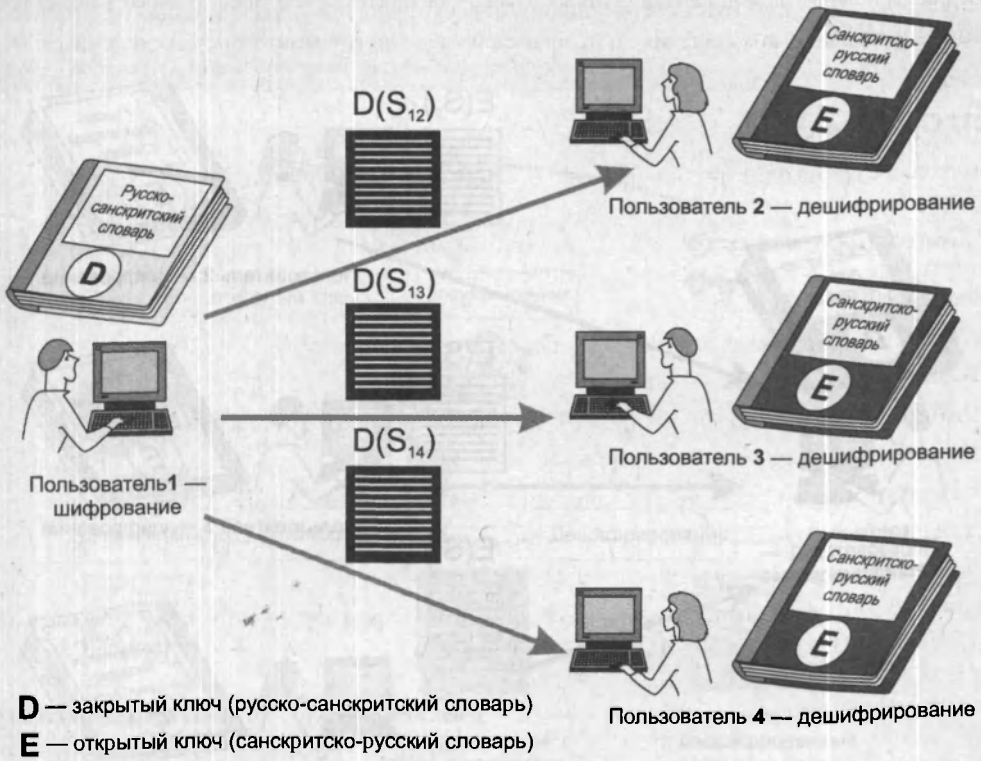


Рис. 24.12. Использование шифрования закрытым ключом для подтверждения авторства

Заметим, что в этом случае сообщения первого пользователя S_{12}, S_{13}, S_{14}, адресованные пользователям 2, 3 и 4, не являются секретными, так как все адресаты обладают одним и тем же открытым ключом, с помощью которого они могут расшифровывать все сообщения, поступающие от пользователя 1.

Для того чтобы в сети все n абонентов имели возможность не только принимать зашифрованные сообщения, но и сами посылать таковые, каждый абонент должен обладать собственной парой ключей E и D. Всего в сети будет $2n$ ключей: n открытых ключей для шифрования и n секретных ключей для дешифрирования. Таким образом решается проблема масштабируемости — квадратичная зависимость количества ключей от числа абонентов в симметричных алгоритмах заменяется линейной зависимостью в несимметричных алгоритмах. Решается и проблема секретной доставки ключа. Злоумышленнику нет смысла стремиться завладеть открытым ключом, поскольку это не дает возможности расшифровывать текст или вычислить закрытый ключ.

Хотя информация об открытом ключе не является секретной, ее нужно защищать от подлогов, чтобы злоумышленник под именем легального пользователя не навязал свой открытый ключ, после чего с помощью своего закрытого ключа он сможет расшифровывать все сообщения, посылаемые легальному пользователю, и отправлять свои сообщения от его имени. Проще всего было бы распространять списки, связывающие имена пользователей с их открытыми ключами, широковещательно путем публикаций в средствах массовой информации (бюллетени, специализированные журналы и т. п.). Однако при таком подходе мы снова, как и в случае с паролями, сталкиваемся с плохой масштабируемостью. Решением проблемы является технология цифровых сертификатов — электронных документов, которые связывают конкретных пользователей с конкретными открытыми ключами.

Алгоритм RSA

В настоящее время одним из наиболее популярных криптоалгоритмов с открытым ключом является криптоалгоритм *RSA*.

В 1978 году трое ученых (Ривест, Шамир и Адлеман) разработали систему шифрования с открытыми ключами **RSA** (Rivest, Shamir, Adleman), полностью отвечающую всем принципам Диффи—Хеллмана. Этот метод состоит в следующем.

1. Случайно выбираются два очень больших простых числа p и q.
2. Вычисляются два произведения $n = p \times q$ и $m = (p - 1) \times (q - 1)$.
3. Выбирается случайное целое число E, не имеющее общих сомножителей с m.
4. Находится D такое, что $DE = 1$ по модулю m.
5. Исходный текст X разбивается на блоки таким образом, чтобы $0 < X < n$.
6. Для шифрования сообщения необходимо вычислить $C = X^E$ по модулю n.
7. Для дешифрирования вычисляется $X = C^D$ по модулю n.

Таким образом, чтобы зашифровать сообщение, необходимо знать пару чисел (E, n), а чтобы расшифровать — пару чисел (D, n). Первая пара — это открытый ключ, а вторая — закрытый.

Зная открытый ключ (E, n), можно вычислить значение закрытого ключа D. Необходимым промежуточным действием в этом преобразовании является нахождение чисел p и q, для чего нужно разложить на простые множители очень большое число n, а на это требуется

очень много времени. Именно с огромной вычислительной сложностью разложения большого числа на простые множители связана высокая криптостойкость алгоритма RSA. В некоторых публикациях приводятся следующие оценки: для того чтобы найти разложение 200-значного числа, понадобится 4 миллиарда лет работы компьютера с быстродействием миллион операций в секунду. Однако следует учесть, что в настоящее время активно ведутся работы по совершенствованию методов разложения больших чисел, поэтому в алгоритме RSA стараются применять числа длиной более 200 десятичных разрядов.

Программная реализация криптоалгоритмов типа RSA значительно сложнее и менее производительна, чем реализации классических криптоалгоритмов типа DES. Вследствие сложности реализации операций модульной арифметики криптоалгоритм RSA обычно используют только для шифрования небольших объемов информации, например для рассылки классических секретных ключей или в алгоритмах цифровой подписи, а основную часть пересылаемой информации шифруют с помощью симметричных алгоритмов.

В табл. 24.1 приведены некоторые сравнительные характеристики классического криптоалгоритма DES и криптоалгоритма RSA

Таблица 24.1. Сравнительные характеристики алгоритмов шифрования

Характеристика	DES	RSA
Скорость шифрования	Высокая	Низкая
Используемая функция шифрования	Перестановка и подстановка	Возведение в степень
Длина ключа	56 бит	Более 500 бит
Наименее затратный криптоанализ (его сложность определяет стойкость алгоритма)	Перебор по всему ключевому пространству	Разложение числа на простые множители
Время генерации ключа	Миллисекунды	Минуты
Тип ключа	Симметричный	Асимметричный

Односторонние функции шифрования

Во многих базовых технологиях безопасности используется еще один прием шифрования — шифрование с помощью **односторонней функции** (one-way function), называемой также **необратимой функцией**, **хэш-функцией** (hash function) или **дайджест-функцией** (digest function).

Эта функция, примененная к шифруемым данным, дает в результате значение, называемое **дайджестом**, которое состоит из фиксированного сравнительно небольшого и не зависящего от длины шифруемого текста числа байтов.

Подчеркнем, знание дайджеста *не позволяет* и даже *не предполагает* восстановления исходных данных. Для чего же нужны односторонние функции шифрования (ОФШ)?

Для ответа на этот вопрос рассмотрим несколько примеров. Пусть требуется обеспечить целостность сообщения, передаваемого по сети. Отправитель и получатель договорились, какую ОФШ и с каким значением параметра — секретного ключа — они будут использовать для решения этой задачи. Прежде чем отправить сообщение, отправитель вычисляет для него дайджест и отправляет его вместе с сообщением адресату (рис. 24.13, *а*). Адресат,

получив данные, применяет ОФШ к переданному в открытом виде исходному сообщению. Если значения вычисленного локально и полученного по сети дайджестов совпадают, значит, содержимое сообщения не было изменено во время передачи.

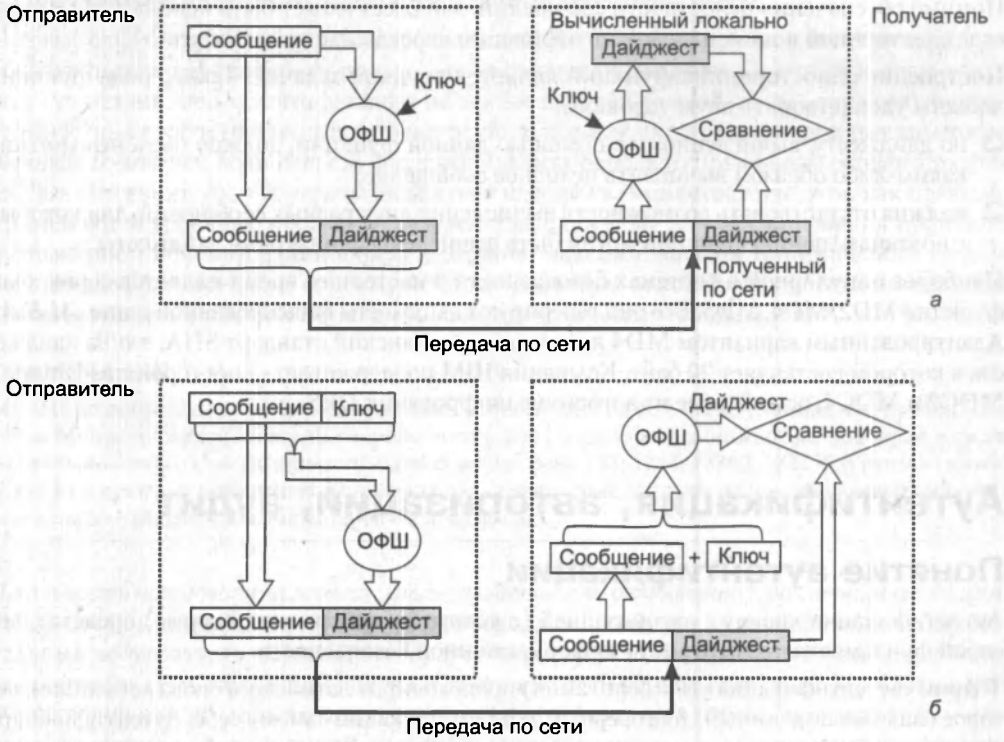


Рис. 24.13. Использование односторонних функций шифрования для контроля целостности

Таким образом, хотя знание дайджеста не дает возможности восстановить исходное сообщение, оно позволяет *проверить целостность данных*.

На первый взгляд кажется, что дайджест является своего рода *контрольной суммой* для исходного сообщения. Однако имеется и существенное отличие. Контрольные суммы применяются тогда, когда нужно обнаружить ошибки, вызванные техническими неполадками, например помехами в линии связи. Это средство не распознает модификацию данных злоумышленником, который, подменив сообщение, может просто добавить к нему заново вычисленную контрольную сумму.

В отличие от контрольной суммы дайджест вычисляется с использованием *параметра* — секретного ключа. Поскольку значение секретного ключа для ОФШ известно только отправителю и получателю, любая модификация исходного сообщения будет немедленно обнаружена.

На рис. 24.13, *б* показан другой вариант использования односторонней функции шифрования для обеспечения целостности данных. Здесь односторонняя функция не имеет параметра-ключа, но зато применяется не просто к сообщению, а к сообщению, дополненному секретным ключом. Получатель извлекает из полученных по сети данных исходное

сообщение, потом дополняет его тем же известным ему секретным ключом и применяет к полученным данным одностороннюю функцию. Результат вычислений сравнивается с полученным по сети дайджестом.

Помимо обеспечения целостности сообщений, дайджест может быть использован в качестве электронной подписи для аутентификации передаваемого документа.

Построение односторонних функций является трудной задачей. Такого рода функции должны удовлетворять двум условиям:

- по дайджесту, вычисленному с помощью данной функции, должно быть невозможно каким-либо образом вычислить исходное сообщение;
- должна отсутствовать возможность вычисления двух разных сообщений, для которых с помощью данной функции могли быть вычислены одинаковые дайджесты.

Наиболее популярной в системах безопасности в настоящее время является серия хэш-функций MD2, MD4, MD5. Все они генерируют дайджесты фиксированной длины 16 байт. Адаптированным вариантом MD4 является американский стандарт SHA, длина дайджеста в котором составляет 20 байт. Компания IBM поддерживает односторонние функции MDC2 и MDC4, основанные на алгоритме шифрования DES.

Аутентификация, авторизации, аудит

Понятие аутентификации

Аутентификация наряду с авторизацией (о которой рассказывается далее) представляет собой фундаментальный атрибут информационной безопасности.

Термин «аутентификация» (authentication) происходит от латинского слова authenticus, которое означает подлинный, достоверный, соответствующий самому себе. Аутентификация, или, другими словами, процедура установления подлинности, может быть применима как к людям, так и другим объектам, в частности к программам, устройствам, документам.

Аутентификация пользователя — это процедура доказательства пользователем того, что он есть тот, за кого себя выдает.

В частности, при выполнении логического входа в защищенную систему пользователь должен пройти процедуру аутентификации, то есть доказать, что именно ему принадлежит введенный им идентификатор (имя пользователя). Аутентификация предотвращает доступ к сети нежелательных лиц и разрешает вход для легальных пользователей.

В процедуре аутентификации участвуют две стороны: одна сторона доказывает свою аутентичность, предъявляя некоторые доказательства, другая сторона — аутентификатор — проверяет эти доказательства и принимает решение. В качестве доказательства аутентичности применяются самые разнообразные приемы:

- аутентифицируемый может продемонстрировать знание некоего общего для обеих сторон секрета: слова (пароля) или факта (даты и места события, прозвища человека и т. п.);
- аутентифицируемый может продемонстрировать, что он владеет неким уникальным предметом (физическим ключом), в качестве которого может выступать, например, электронная магнитная карта;

- аутентифицируемый может доказать свою идентичность, используя собственные биохарактеристики: рисунок радужной оболочки глаза или отпечатки пальцев, которые предварительно были занесены в базу данных аутентификатора.

Сетевые службы аутентификации строятся на основе всех этих приемов, но чаще всего для доказательства идентичности пользователя применяют **пароли**. Простота и логическая ясность механизмов аутентификации на основе паролей в какой-то степени компенсирует известные слабости паролей. Это, во-первых, возможность раскрытия и разгадывания паролей, во-вторых, возможность «подслушивания» пароля путем анализа сетевого трафика. Для снижения уровня угрозы раскрытия паролей администраторы сети, как правило, применяют встроенные программные средства, служащие для формирования политики назначения и использования паролей: задание максимального и минимального сроков действия пароля, хранение списка уже использованных паролей, управление поведением системы после нескольких неудачных попыток логического входа и т. п.

ПРИМЕЧАНИЕ

Многие пользователи пренебрегают угрозами, которые несут в себе легко угадываемые пароли. Так, червь Mumu, поразивший компьютерные сети в 2003 году, искал свои жертвы, подбирая пароли из очень короткого списка: password, passwd, admin, pass, 123, 1234, 12345, 123456 и пустая строка. Такая на удивление примитивная стратегия дала прекрасные (с точки зрения атакующей стороны) результаты — множество компьютеров было взломано.

Легальность пользователя может устанавливаться по отношению к различным системам. Так, работая в сети, пользователь может проходить процедуру аутентификации и как локальный пользователь, который претендует на ресурсы только данного компьютера, и как пользователь сети, желающий получить доступ ко всем сетевым ресурсам. При локальной аутентификации пользователь вводит свои идентификатор и пароль, которые автономно обрабатываются операционной системой, установленной на данном компьютере. При логическом входе в сеть данные о пользователе (идентификатор и пароль) передаются на сервер, который хранит учетные записи всех пользователей сети. Однако такая упрощенная схема имеет большой изъян — при передаче пароля с клиентского компьютера на сервер, выполняющий процедуру аутентификации, этот пароль может быть перехвачен злоумышленником. Поэтому применяются разные приемы, чтобы избежать передачи пароля по сети в незащищенном виде.

Аутентификация, в процессе которой используются методы шифрования, а аутентификационная информация не передается по сети, называется **строгой**.

Многие приложения имеют собственные средства определения, является ли пользователь законным. И тогда пользователю приходится проходить дополнительные этапы проверки.

Как уже отмечалось, в качестве объектов, требующих аутентификации, могут выступать не только пользователи, но и различные приложения, устройства, текстовая и другая информация.

Так, пользователь, обращающийся с запросом к корпоративному веб-серверу, должен доказать ему свою легальность, но он также должен убедиться сам, что ведет диалог действительно с веб-сервером своего предприятия. Другими словами, сервер и клиент должны пройти процедуру взаимной аутентификации. Здесь мы имеем дело с *аутентификацией на уровне приложений*.

При установлении сеанса связи между двумя устройствами также часто предусматриваются процедуры взаимной *аутентификации устройств* на более низком, канальном, уровне (см. далее раздел «Строгая аутентификация на основе многоразового пароля в протоколе CHAP»).

Аутентификация данных означает доказательство целостности этих данных, а также то, что они поступили именно от того человека, который объявил об этом. Для этого используется механизм **электронной подписи**. Ранее мы уже узнали, как используется для аутентификации данных несимметричное шифрование.

Авторизация доступа

Термин авторизация (authorization) происходит от латинского слова auctoritas, показывающее уровень престижа человека в Древнем Риме и соответствующие этому уровню привилегии.

Авторизация — это процедура контроля доступа легальных пользователей к ресурсам системы и предоставление каждому из них именно тех прав, которые ему были определены администратором.

В отличие от аутентификации, которая позволяет распознать легальных и нелегальных пользователей, авторизация имеет дело только с *легальными* пользователями, успешно прошедшими процедуру аутентификации. Помимо предоставления пользователям прав доступа к каталогам, файлам и принтерам, средства авторизации могут контролировать возможность выполнения пользователями различных системных функций, таких как локальный доступ к серверу, установка системного времени, создание резервных копий данных, выключение сервера и т. п.

Средства авторизации наделяют пользователя сети правами выполнять определенные действия по отношению к определенным ресурсам. Для этого могут применяться различные формы предоставления правил доступа, которые часто делят на два класса:

- **Избирательный доступ** наиболее широко используется в компьютерных сетях. При этом подходе определенные операции с определенным ресурсом разрешаются или запрещаются пользователям или группам пользователей, *явно указанным своими идентификаторами*, например: «пользователю User_T разрешено читать и записывать в файл File1».
- **Мандатный подход** к определению прав доступа заключается в том, что вся информация делится на уровни в зависимости от степени секретности, а все пользователи сети также делятся на группы, образующие иерархию в соответствии с *уровнем допуска* к этой информации. Такой подход позволяет классифицировать данные на информацию для служебного пользования, а также секретную и совершенно секретную информацию. Пользователи этой информации в зависимости от определенного для них статуса получают разные формы допуска: первую, вторую или третью. В отличие от систем с избирательными правами доступа, в системах с мандатным подходом пользователи в принципе не имеют возможности изменить уровень доступности информации. Например, пользователь более высокого уровня не может разрешить читать данные из своего файла пользователю, относящемуся к более низкому уровню. Отсюда видно, что мандатный подход является более строгим.

Процедуры авторизации часто совмещаются с процедурами аутентификации и реализуются одними и теми же программными средствами, которые могут встраиваться в операционную систему или приложение, а также поставляться в виде отдельных программных продуктов. При этом программные системы аутентификации и авторизации могут строиться на базе двух схем:

- *Централизованная схема, базирующаяся на сервере.* В этой схеме сервер управляет процессом предоставления ресурсов сети пользователю. Главная цель таких систем — реализовать «принцип единого входа». В соответствии с централизованной схемой пользователь один раз логически входит в сеть и получает на все время работы некоторый набор разрешений по доступу к различным ресурсам сети. Система Kerberos[1] с ее сервером безопасности и архитектурой клиент-сервер, а также более современная система Shibboleth, построенная в той же архитектуре, являются наиболее известными системами этого типа. Системы TACACS и RADIUS, часто применяемые совместно с системами удаленного доступа, также реализуют этот подход.
- *Децентрализованная схема, базирующаяся на рабочих станциях.* При этом подходе средства авторизации работают на каждой машине. Администратор должен отслеживать работу механизмов безопасности каждого отдельного приложения — электронной почты, справочной службы, локальных баз данных и т. п.

Подчеркнем, что системы аутентификации и авторизации совместно решают одну задачу — обеспечение контроля доступа, поэтому к ним необходимо предъявлять одинаковый уровень требований. Ненадежность одного звена здесь не может быть компенсирована надежностью другого.

Аудит

Аудит (auditing) — это набор процедур мониторинга и учета всех событий, представляющих потенциальную угрозу для безопасности системы.

Аудит позволяет «шпионить» за выбранными объектами и выдавать сообщения тревоги, когда, например, какой-либо рядовой пользователь попытается прочитать или модифицировать системный файл. Если кто-то пытается выполнить действия, выбранные системой безопасности для мониторинга, то система аудита пишет сообщение в журнал регистрации, идентифицируя пользователя. Системный менеджер может готовить отчеты безопасности, которые содержат информацию из журнала регистрации. Для «сверхбезопасных» систем предусматриваются аудио- и видеосигналы тревоги, устанавливаемые на машинах администраторов, отвечающих за безопасность.

Поскольку никакая система безопасности не гарантирует защиту на уровне 100 %, последним рубежом в борьбе с нарушениями оказывается система аудита. Действительно, после того как злоумышленнику удалось провести успешную атаку, пострадавшей стороне не остается ничего другого, как обратиться к службе аудита. Если при настройке службы аудита были правильно заданы события, которые требуется отслеживать, то подробный анализ записей в журнале может дать много полезной информации. Эта информация, возможно, позволит найти злоумышленника или, по крайней мере, предотвратить повто-

[1] Детали о системе Kerberos см. в книге авторов «Сетевые операционные системы».

рение подобных атак путем устранения уязвимых мест в системе защиты. Функции аудита встраиваются в различные средства обеспечения безопасности: сетевые экраны, системы обнаружения вторжений, антивирусные системы, сетевые мониторы.

Строгая аутентификация на основе многоразового пароля в протоколе CHAP

Протокол аутентификации по квитированию вызова (Challenge Handshake Authentication Protocol, CHAP) входит в семейство протоколов PPP. В этом протоколе предусмотрено 4 типа сообщений: *Success* (успех), *Challenge* (вызов), *Response* (ответ), *Failure* (ошибка).

Этот протокол используется, например, при аутентификации удаленных пользователей, подключенных к Интернету по коммутируемому каналу. Здесь аутентификатором является сервер провайдера, а аутентифицируемым — клиентский компьютер (рис. 24.14). При заключении договора клиент получает от провайдера пароль (пусть, например, это будет слово parol). Этот пароль хранится в базе данных провайдера в виде дайджеста $Z = d(\text{parol})$, полученного путем применения к паролю односторонней хэш-функции MD5.

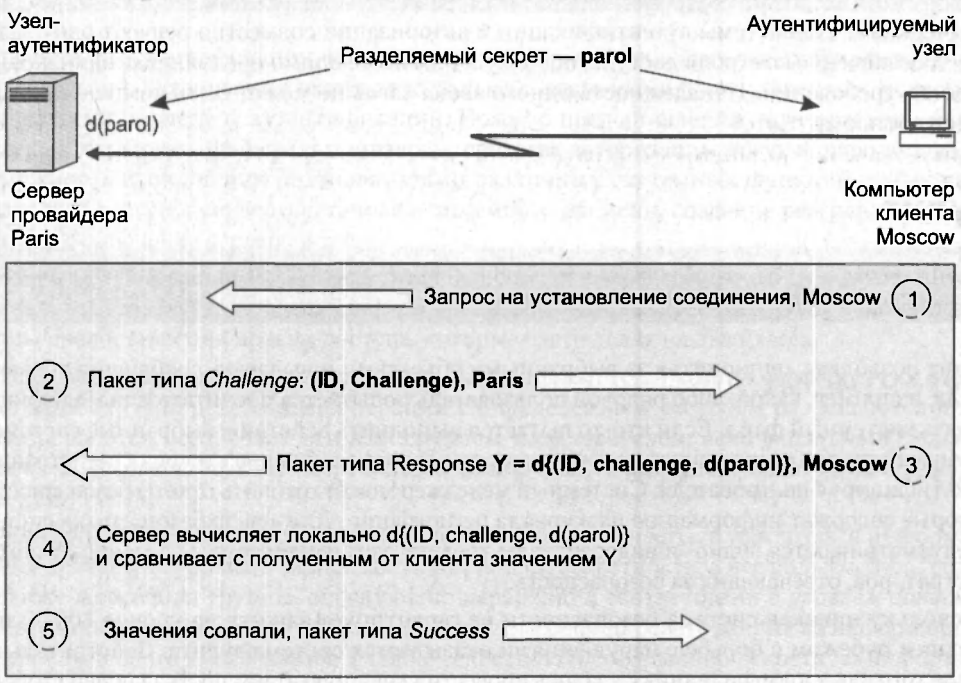


Рис. 24.14. Аутентификация по протоколу CHAP

Аутентификация выполняется в следующей последовательности.

1. Пользователь-клиент активизирует программу (например, программу дозвона) удаленного доступа к серверу провайдера, вводя имя и назначенный ему пароль. Имя (на рисунке это «Moscow») передается по сети провайдеру в составе запроса на соединение,

но пароль не передается в сеть ни в каком виде. То есть здесь мы имеем дело со *строгой* аутентификацией.

2. Сервер провайдера, получив запрос от клиента, генерирует псевдослучайное слово-вызов (пусть это будет слово «challenge») и передает его клиенту вместе со значением, идентифицирующем сообщение в рамках данного сеанса (ID), и собственным именем (здесь «Paris»). Это сообщение типа *Challenge*. (Для защиты от перехвата ответа аутентификатор должен использовать разные значения слова-вызова при каждой процедуре аутентификации.)
3. Программа клиента, получив этот пакет, извлекает из него слово-вызов, добавляет к нему идентификатор и вычисленный локально дайджест $Z = d(\text{parol})$, а затем вычисляет с помощью все той же функции MD5 дайджест $Y = d\{(\text{ID}, \text{challenge}, d(\text{parol})\}$ от всех этих трех значений. Результат клиент посылает серверу провайдера в пакете *Response*.
4. Сервер провайдера сравнивает полученный по сети дайджест Y с тем значением, которое он получил, локально применив ту же хэш-функцию к набору аналогичных компонентов, хранящихся в его памяти.
5. Если результаты совпадают, то аутентификация считается успешной и аутентификатор посылает партнеру пакет *Success*.

Аналогичный алгоритм аутентификации применяется в семействе ОС Windows. Там многоразовые пароли пользователей также хранятся в базе данных сервера в виде дайджестов, а по сети в открытом виде передается только слово-вызов. Кажется, что такой способ хранения паролей надежно защищает их от злоумышленника, даже если он сможет получить к ним доступ. Действительно, ведь даже теоретически нельзя восстановить исходное значение по дайджесту. Однако создатель первого червя Роберт Моррис решил эту проблему. Он разработал довольно простую программу, которая генерировала возможные варианты паролей, как используя слова из словаря, так и путем последовательного перебора символов. Для каждого сгенерированного слова вычислялся дайджест и сравнивался с дайджестами из файла паролей. Удивительно, но такая стратегия оказалась весьма эффективной, и хакеру удалось завладеть несколькими паролями.

Аутентификация на основе одноразового пароля

Алгоритмы аутентификации, основанные на многоразовых паролях, не очень надежны. Пароли можно подсмотреть, разгадать или просто украсть. Более надежными оказываются схемы с **одноразовыми паролями**. К тому же одноразовые пароли намного дешевле и проще биометрических систем аутентификации, таких как сканеры сетчатки глаза или отпечатков пальцев. Все это делает системы, основанные на одноразовых паролях, очень перспективными. Следует иметь в виду, что, как правило, системы аутентификации на основе одноразовых паролей рассчитаны на проверку только *удаленных*, а не локальных пользователей.

Генерация одноразовых паролей может выполняться либо программно, либо аппаратно. Аппаратные реализации систем доступа на основе одноразовых паролей называют **аппаратными ключами**. Они представляют собой миниатюрные устройства со встроенным микропроцессором, похожие либо на обычные пластиковые карточки, используемые для доступа к банкоматам, либо на карманные калькуляторы, имеющие клавиатуру и маленькое дисплейное окно (рис. 24.15). Аппаратные ключи могут быть также реализованы в виде присоединяемого к разъему компьютера устройства.

Рис. 24.15. Аппаратный ключ, который используют клиенты банка Barclays для доступа к своим счетам

Существуют и программные реализации средств аутентификации на основе одноразовых паролей — **программные ключи**. Программные ключи размещаются на сменном магнитном носителе в виде обычной программы, важной частью которой является генератор одноразовых паролей.

Независимо от того, какую реализацию системы аутентификации на основе одноразовых паролей выбирает пользователь, он, как и в системах аутентификации с применением многоразовых паролей, сообщает системе свой идентификатор, однако вместо того чтобы вводить каждый раз один и тот же пароль, он указывает последовательность цифр, сообщаемую ему аппаратным или программным ключом. Через определенный небольшой период времени генерируется другая последовательность — новый пароль. Сервер аутентификации проверяет введенную последовательность и разрешает пользователю осуществить логический вход. Сервер аутентификации может представлять собой отдельное устройство, выделенный компьютер или же программу, выполняемую на обычном сервере.

Рассмотрим схему использования аппаратных ключей, в основе которой лежит *синхронизация по времени*. Этот популярный алгоритм аутентификации был разработан компанией Security Dynamics.

Идея метода состоит в том, что аппаратный ключ и аутентифицирующий сервер вычисляют некоторое значение по одному и тому же алгоритму. Алгоритм имеет два параметра:

- разделяемый секретный ключ, представляющий собой 64-разрядное число, уникально назначаемое каждому пользователю и хранящееся как в аппаратном ключе, так и в базе данных сервера аутентификации;
- значение текущего времени.

Если вычисленные значения совпадают, то аутентификация считается успешной.

Итак, пусть удаленный пользователь пытается совершить логический вход в систему с персонального компьютера (рис. 24.16). Аутентифицирующая программа предлагает ему ввести его личный персональный номер (PIN), состоящий из четырех десятичных цифр, а также 6 цифр случайного числа, отображаемого в тот момент на дисплее аппаратного ключа. На основе PIN-кода сервер извлекает из базы данных информацию о пользователе, а именно — его секретный ключ. Затем сервер выполняет вычисления по тому же алгоритму, который заложен в аппаратном ключе, используя в качестве параметров секретный ключ и значение текущего времени, проверяя, совпадает ли сгенерированное число с числом, которое ввел пользователь. Если они совпадают, то пользователю разрешается логический вход.

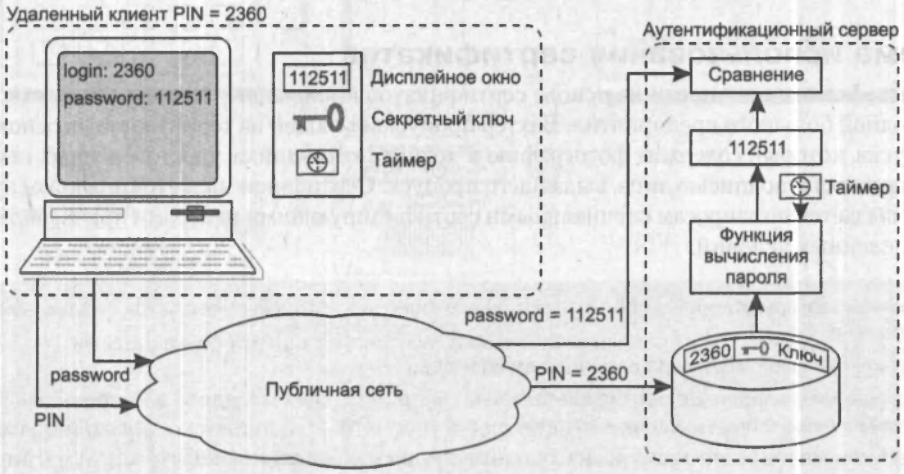


Рис. 24.16. Аутентификация, основанная на временно́й синхронизации

Потенциальной проблемой этой схемы является временна́я синхронизация сервера и аппаратного ключа (ясно, что вопрос согласования часовых поясов решается просто). Гораздо сложнее обстоит дело с постепенным рассогласованием внутренних часов сервера и аппаратного ключа, тем более что потенциально аппаратный ключ может работать несколько лет. Компания Security Dynamics решает эту проблему двумя способами. Во-первых, при производстве аппаратного ключа измеряется отклонение частоты его таймера от номинала. Далее эта величина учитывается в виде параметра алгоритма сервера. Во-вторых, сервер отслеживает коды, генерируемые конкретным аппаратным ключом, и если таймер данного ключа постоянно спешит или отстает, то сервер динамически подстраивается под него.

Существует еще одна проблема, связанная со схемой временно́й синхронизации. Одноразовый пароль, генерируемый аппаратным ключом, действителен в течение некоторого интервала времени (от нескольких десятков секунд до нескольких десятков минут), то есть в течение этого времени одноразовый пароль, в сущности, является многоразовым. Поэтому теоретически возможно, что очень проворный хакер сможет перехватить PIN-код и одноразовый пароль с тем, чтобы также получить доступ в сеть в течение этого интервала.

Аутентификация на основе сертификатов

Аутентификация с применением цифровых сертификатов является альтернативой применению паролей и представляется естественным решением в условиях, когда число пользо-

вателей сети (пусть и потенциальных) измеряется миллионами. В таких обстоятельствах процедура предварительной регистрации пользователей, связанная с назначением и хранением их паролей, становится крайне обременительной, опасной, а иногда и просто нереализуемой. При наличии сертификатов сеть, которая дает пользователю доступ к своим ресурсам, не хранит никакой информации о своих пользователях — они ее предоставляют сами в своих запросах в виде сертификатов, удостоверяющих личность пользователей. Сертификаты выдаются специальными уполномоченными организациями — **центрами сертификации** (Certificate Authority, CA). Поэтому задача хранения секретной информации (закрытых ключей) возлагается на самих пользователей, что делает это решение гораздо более масштабируемым, чем вариант с централизованной базой паролей.

Схема использования сертификатов

Аутентификация личности на основе сертификатов происходит примерно так же, как на проходной большого предприятия. Вахтер пропускает людей на территорию на основании пропуска, который содержит фотографию и подпись сотрудника, удостоверенных печатью предприятия и подписью лица, выдавшего пропуск. Сертификат является аналогом пропуска и выдается по запросам специальными сертифицирующими центрами при выполнении определенных условий.

Сертификат представляет собой электронную форму, в которой содержится следующая информация:

- открытый ключ владельца данного сертификата;
- сведения о владельце сертификата, такие, например, как имя, адрес электронной почты, наименование организации, в которой он работает и т. п.;
- наименование сертифицирующей организации, выдавшей данный сертификат;
- электронная подпись сертифицирующей организации, то есть зашифрованные закрытым ключом этой организации данные, содержащиеся в сертификате.

Использование сертификатов основано на предположении, что сертифицирующих организаций немного и их открытые ключи широко доступны, например, из публикаций в журналах.

Когда пользователь хочет подтвердить свою личность, он предъявляет свой сертификат в двух формах: открытой (то есть такой, в которой он получил его в сертифицирующей организации) и зашифрованной с применением своего закрытого ключа (рис. 24.17). Сторона, проводящая аутентификацию, берет из незашифрованного сертификата открытый ключ пользователя и расшифровывает с его помощью зашифрованный сертификат. Совпадение результата с открытым сертификатом подтверждает, что предъявитель действительно является владельцем закрытого ключа, соответствующего указанному открытому.

Затем с помощью известного открытого ключа указанной в сертификате организации проводится расшифровка подписи этой организации в сертификате. Если в результате получается тот же сертификат с тем же именем пользователя и его открытым ключом, значит, он действительно прошел регистрацию в сертификационном центре, является тем, за кого себя выдает, и указанный в сертификате открытый ключ действительно принадлежит ему.

Сертификаты можно использовать не только для аутентификации, но и для предоставления избирательных прав доступа. Для этого в сертификат могут вводиться дополнительные поля, в которых указывается принадлежность его владельцев к той или иной категории пользователей. Эта категория назначается сертифицирующей организацией в зависимо-

сти от условий, на которых выдается сертификат. Например, организация, поставляющая через Интернет на коммерческой основе информацию, может выдавать сертификаты определенной категории пользователям, оплатившим годовую подписку на некоторый бюллетень, тогда веб-сервер будет предоставлять доступ к страницам бюллетеня только пользователям, предъявившим сертификат данной категории.

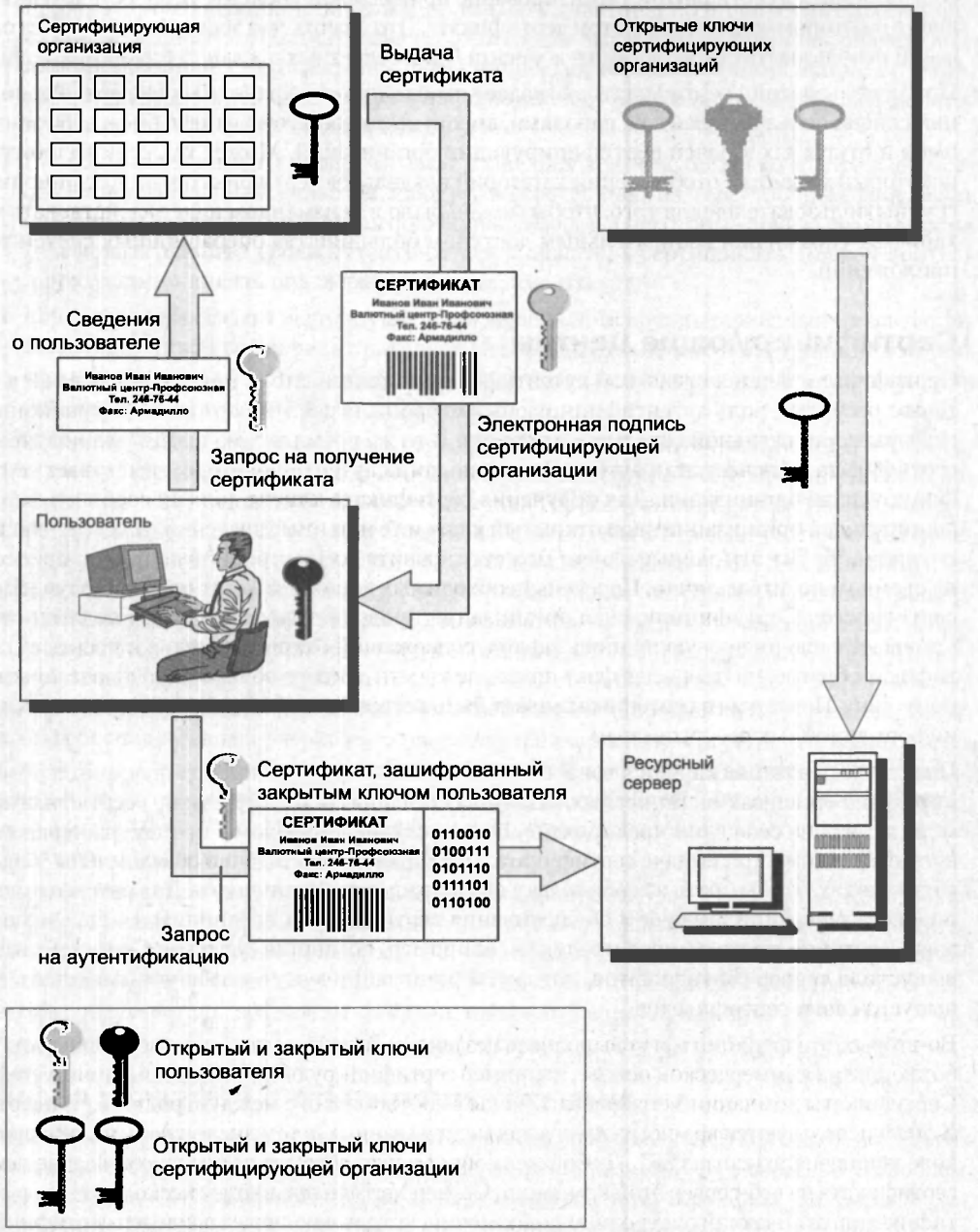


Рис. 24.17. Аутентификация пользователей на основе сертификатов

Подчеркнем тесную связь открытых ключей с сертификатами. Сертификат является удостоверением не только личности, но и принадлежности открытого ключа. Цифровой сертификат устанавливает и гарантирует соответствие между открытым ключом и его владельцем. Это предотвращает угрозу подмены открытого ключа. Если некоторый абонент *А* получает по сети сертификат от абонента *Б*, то он может быть уверен, что открытый ключ, содержащийся в сертификате, гарантированно принадлежит абоненту *Б*, адрес и другие сведения о котором содержатся в этом сертификате. Это значит, что абонент *А* может без опасений использовать открытый ключ абонента *Б* для секретных посланий в адрес последнего.

При использовании сертификатов отпадает необходимость хранить на серверах корпораций списки пользователей с их паролями, вместо этого достаточно иметь на сервере список имен и открытых ключей сертифицирующих организаций. Может также понадобиться некоторый механизм отображений категорий владельцев сертификатов на традиционные группы пользователей для того, чтобы можно было в неизменном виде задействовать механизмы управления избирательным доступом большинства операционных систем или приложений.

Сертифицирующие центры

Сертификат является средством аутентификации пользователя при его обращении к сетевым ресурсам, роль аутентифицирующей стороны играют при этом информационные серверы корпоративной сети или Интернета. В то же время и сама процедура получения сертификата включает этап аутентификации, когда аутентификатором выступает сертифицирующая организация. Для получения сертификата клиент должен сообщить сертифицирующей организации свой открытый ключ и те или иные сведения, удостоверяющие его личность. Все эти данные клиент может отправить по электронной почте или принести на съемном носителе лично. Перечень необходимых данных зависит от типа получаемого сертификата. Сертифицирующая организация проверяет доказательства подлинности, помещает свою цифровую подпись в файл, содержащий открытый ключ, и посылает сертификат обратно, подтверждая факт принадлежности данного конкретного ключа конкретному лицу. После этого сертификат может быть встроен в любой запрос на использование информационных ресурсов сети.

Практически важным является вопрос о том, кто имеет право выполнять функции сертифицирующей организации. Во-первых, задачу обеспечения своих сотрудников сертификатами может взять на себя само предприятие. В этом случае упрощается процедура первичной аутентификации при выдаче сертификата. Предприятия достаточно осведомлены о своих сотрудниках, чтобы брать на себя задачу подтверждения их личности. Для автоматизации процесса генерации, выдачи и обслуживания сертификатов предприятия могут использовать готовые программные продукты, например, компания Netscape Communications выпустила сервер сертификатов, который организации могут у себя устанавливать для выпуска своих сертификатов.

Во-вторых, эти функции могут выполнять независимые центры по выдаче сертификатов, работающие на коммерческой основе, например сертифицирующий центр компании Verisign. Сертификаты компании Verisign выполнены в соответствии с международным стандартом X.509 и используются во многих продуктах защиты данных, в том числе в популярном протоколе защищенного канала SSL. Любой желающий может обратиться с запросом на получение сертификата на веб-сервер этой компании. Сервер Verisign предлагает несколько типов сертификатов, отличающихся уровнем полномочий, которые получает владелец сертификата.

- *Сертификаты класса 1* предоставляют пользователю самый низкий уровень полномочий. Они могут применяться при отправке и получении шифрованной электронной почты через Интернет. Чтобы получить сертификат этого класса, пользователь должен сообщить серверу Verisign свой адрес электронной почты или свое уникальное имя.
- *Сертификаты класса 2* дают возможность его владельцу пользоваться внутрикорпоративной электронной почтой и принимать участие в подписных интерактивных службах. Чтобы получить сертификат этого более высокого уровня, пользователь должен организовать подтверждение своей личности сторонним лицом, например своим работодателем. Такой сертификат с информацией от работодателя может эффективно применяться при деловой переписке.
- *Сертификаты класса 3* предоставляют владельцу все те возможности, которые имеет обладатель сертификата класса 2, плюс возможность участия в электронных банковских операциях, электронных сделках по покупке товаров и некоторые другие возможности. Для доказательства своей аутентичности соискатель сертификата должен явиться лично и предоставить подтверждающие документы.
- *Сертификаты класса 4* используются при выполнении крупных финансовых операций. Поскольку такой сертификат наделяет владельца самым высоким уровнем доверия, сертифицирующий центр Verisign проводит тщательное изучение частного лица или организации, запрашивающей сертификат.

Механизм получения пользователем сертификата хорошо автоматизируется в сети в модели клиент-сервер, когда браузер исполняет роль клиента, а в сертифицирующей организации установлен специальный сервер выдачи сертификатов. Браузер генерирует для пользователя пару ключей, оставляет закрытый ключ у себя и передает частично заполненную форму сертификата серверу. Для того чтобы неподписанный еще сертификат нельзя было подменить при передаче по сети, браузер ставит свою электронную подпись, зашифровывая сертификат выработанным закрытым ключом. Сервер сертификатов подписывает полученный сертификат, фиксирует его в своей базе данных и возвращает его каким-либо способом владельцу. Очевидно, что при этом может выполняться еще и неформальная процедура подтверждения пользователем своей личности и права на получение сертификата, требующая участия оператора сервера сертификатов. Это могут быть доказательства оплаты услуги, доказательства принадлежности к той или иной организации — все случаи жизни предусмотреть и автоматизировать нельзя. После получения сертификата браузер сохраняет его вместе с закрытым ключом и использует при аутентификации на тех серверах, которые поддерживают такой процесс.

В настоящее время существует большое количество протоколов и продуктов, использующих сертификаты. Например, компания Microsoft реализовала поддержку сертификатов и в своем браузере Internet Explorer, и в сервере Internet Information Server, разработала собственный сервер сертификатов, а также продукты, которые позволяют хранить сертификаты пользователя, его закрытые ключи и пароли защищенным образом.

Инфраструктура с открытыми ключами

Несмотря на активное использование технологии цифровых сертификатов во многих системах безопасности, эта технология еще не решила целый ряд серьезных проблем. Это, прежде всего, поддержание базы данных о выпущенных сертификатах. Сертификат выдается не навсегда, а на некоторый вполне определенный срок. По истечении срока

годности сертификат должен либо обновляться, либо аннулироваться. Кроме того, необходимо предусмотреть возможность досрочного прекращения полномочий сертификата. Все заинтересованные участники информационного процесса должны быть вовремя оповещены о том, что некоторый сертификат уже недействителен. Для этого сертифицирующая организация должна оперативно поддерживать список аннулированных сертификатов.

Имеется также ряд проблем, связанных с тем, что сертифицирующие организации существуют не в единственном числе. Все они выпускают сертификаты, но даже если эти сертификаты соответствуют единому стандарту (сейчас это, как правило, стандарт X.509), все равно остаются нерешенными многие вопросы. Все ли сертифицирующие центры заслуживают доверия? Каким образом можно проверить полномочия того или иного сертифицирующего центра? Можно ли создать иерархию сертифицирующих центров, когда сертифицирующий центр, стоящий выше, мог бы сертифицировать центры, расположенные ниже в иерархии? Как организовать совместное использование сертификатов, выпущенных разными сертифицирующими организациями?

Для решения этих и многих других проблем, возникающих в системах, использующих технологии шифрования с открытыми ключами, оказывается необходимым комплекс программных средств и методик, называемый **инфраструктурой с открытыми ключами** (Public Key Infrastructure, PKI). Информационные системы больших предприятий нуждаются в специальных средствах администрирования и управления цифровыми сертификатами, парами открытых/закрытых ключей, а также приложениями, функционирующими в среде с открытыми ключами,

В настоящее время любой пользователь имеет возможность, загрузив широко доступное программное обеспечение, абсолютно бесконтрольно сгенерировать себе пару открытый/закрытый ключ. Затем он может также совершенно независимо от администрации вести шифрованную переписку со своими внешними абонентами. Такая «свобода» пользователя часто не соответствует принятой на предприятии политике безопасности. Для более надежной защиты корпоративной информации желательно реализовать централизованную службу генерации и распределения ключей. Для администрации предприятия важно иметь возможность получить копии закрытых ключей каждого пользователя сети, чтобы в случае увольнения пользователя или потери пользователем его закрытого ключа сохранить доступ к зашифрованным данным этого пользователя. В противном случае резко ухудшается одна из трех характеристик безопасной системы — доступность данных.

Процедура, позволяющая получать копии закрытых ключей, называется **восстановлением ключей**. Вопрос, включать ли в продукты безопасности средства восстановления ключей, в последние годы приобрел политический оттенок. В США прошли бурные дебаты, тему которых можно примерно сформулировать так: обладает ли правительство правом доступа к любой частной информации при условии, что на это есть постановление суда?

И хотя в такой широкой постановке проблема восстановления ключей все еще не решена, необходимость включения средств восстановления в корпоративные продукты ни у кого сомнений не вызывает. Принцип доступности данных не должен нарушаться из-за волюнтаризма сотрудников, монопольно владеющих своими закрытыми ключами. Ключ может быть восстановлен при выполнении некоторых условий, которые должны быть четко определены в политике безопасности предприятия.

Как только принимается решение о включении в систему безопасности средств восстановления, возникает вопрос, как же быть с надежностью защиты данных, как убедить поль-

зователя в том, что его закрытый ключ не употребляется с какими-либо другими целями, не имеющими отношения к резервированию? Некоторую уверенность в секретности хранения закрытых ключей может дать технология **депонирования ключей**. Депонирование ключей — это предоставление закрытых ключей на хранение третьей стороне, надежность которой не вызывает сомнений. Этой третьей стороной может быть правительственная организация или группа уполномоченных на это сотрудников предприятия, которым оказывается полное доверие.

Аутентификация информации

Под **аутентификацией информации** в компьютерных системах понимают установление подлинности полученных по сети данных исключительно на основе информации, содержащейся в полученном сообщении.

Если конечной целью шифрования информации является защита от несанкционированного ознакомления с этой информацией, то конечной целью аутентификации информации является *защита участников информационного обмена от навязывания ложной информации*. Концепция аутентификации в широком смысле предусматривает установление подлинности информации как при наличии взаимного доверия между участниками обмена, так и при его отсутствии.

В компьютерных системах выделяют два вида аутентификации информации:

- аутентификация хранящихся массивов данных и программ — установление факта того, что данные не подвергались модификации;
- аутентификация сообщений — установление подлинности полученного сообщения, в том числе решение вопроса об авторстве этого сообщения и установление факта приема.

Цифровая подпись

Для решения задачи аутентификации информации используется концепция **цифровой**, или **электронной**, **подписи**. Согласно терминологии, утвержденной Международной организацией по стандартизации (ISO), под термином «цифровая подпись» понимаются методы, позволяющие устанавливать подлинность автора сообщения (документа) при возникновении спора относительно авторства. Основная область применения цифровой подписи — это финансовые документы, сопровождающие электронные сделки, документы, фиксирующие международные договоренности и т. п.

До настоящего времени чаще всего для построения схемы цифровой подписи использовался алгоритм RSA. Как уже отмечалось (см. раздел «Алгоритм RSA»), в основе этого алгоритма лежит концепция Диффи—Хеллмана. Она заключается в том, что каждый пользователь сети имеет свой закрытый ключ, необходимый для формирования подписи, а соответствующий этому секретному ключу открытый ключ, предназначенный для проверки подписи, известен всем другим пользователям сети.

На рис. 24.18 показана схема формирования цифровой подписи по алгоритму RSA. Подписанное сообщение состоит из двух частей: незашифрованной части, в которой содержится исходный текст T, и зашифрованной части, представляющей собой цифровую подпись. Цифровая подпись S вычисляется с использованием закрытого ключа (D, n) по формуле: $S = T^D \bmod n$.

Рис. 24.18. Схема формирования цифровой подписи по алгоритму RSA

Сообщение посылается в виде пары (T, S). Каждый пользователь, имеющий соответствующий открытый ключ (E, n), получив сообщение, отделяет открытую часть T, расшифровывает цифровую подпись S и проверяет равенство: $T = S^E \bmod n$.

Если результат расшифровки цифровой подписи совпадает с открытой частью сообщения, считается, что документ подлинный, не претерпел никаких изменений в процессе передачи, а автором его является именно тот человек, который передал свой открытый ключ получателю. Если сообщение снабжено цифровой подписью, то получатель может быть уверен, что оно не было изменено или подделано по пути. Такие схемы аутентификации называются асимметричными. К недостаткам данного алгоритма можно отнести то, что длина подписи в этом случае равна длине сообщения, что не всегда удобно.

Если помимо проверки целостности документа, обеспечиваемой цифровой подписью, надо обеспечить его *конфиденциальность*, то после применения к тексту цифровой подписи выполняют шифрование и исходного текста, и цифровой подписи (рис. 24.19).

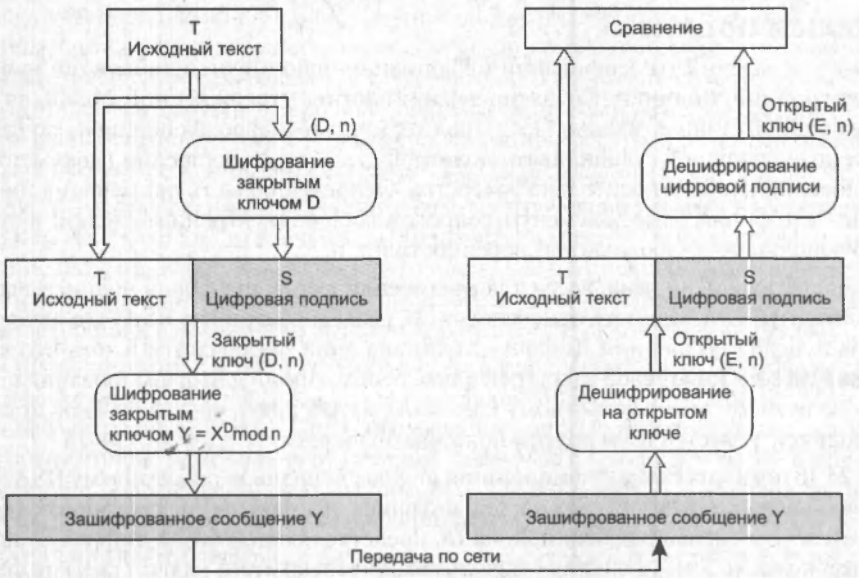


Рис. 24.19. Обеспечение конфиденциальности документа с цифровой подписью

Другие методы цифровой подписи основаны на формировании соответствующей сообщению контрольной комбинации с помощью симметричных алгоритмов типа DES. Учитывая более высокую производительность алгоритма DES по сравнению с RSA, он более эффективен для подтверждения аутентичности больших объемов информации. А для коротких сообщений типа платежных поручений или квитанций подтверждения приема, наверное, лучше подходит алгоритм RSA.

Аутентификация программных кодов

Компания Microsoft разработала средства для доказательства аутентичности программных кодов, распространяемых через Интернет. Пользователю важно иметь доказательства, что программа, которую он загрузил с какого-либо сервера, действительно содержит коды, разработанные определенной компанией. Протоколы защищенного канала (см. далее) типа SSL помочь здесь не могут, так как позволяют удостоверить только аутентичность сервера. Суть технологии **аутентикода** (authenticode), разработанной Microsoft, состоит в следующем.

Организация, желающая подтвердить свое авторство на программу, должна встроить в распространяемый код так называемый подписывающий блок (рис. 24.20). Этот блок состоит из двух частей. Первая часть — сертификат этой организации, полученный обычным образом от какого-либо сертифицирующего центра. Вторую часть образует зашифрованный дайджест, полученный в результате применения односторонней функции к распространяемому коду. Шифрование дайджеста выполняется с помощью закрытого ключа организации.

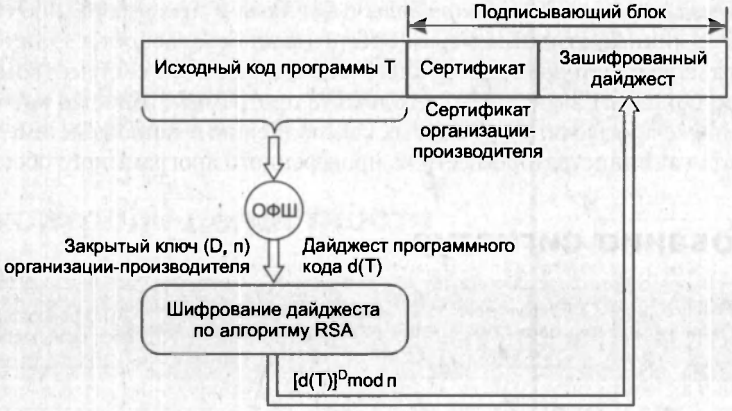


Рис. 24.20. Схема получения аутентикода

Антивирусная защита

Антивирусная защита используется для профилактики и диагностики вирусного заражения, а также для восстановления работоспособности пораженных вирусами информационных систем.

Термин «вирусы» толкуется здесь расширенно — это не только собственно вирусы, но и другие разновидности вредоносных программ, такие как черви, троянские и шпионские программы.

Профилактика заключается в проверке файлов на присутствие вирусов перед их загрузкой на защищаемый компьютер и тем более перед их выполнением на этом компьютере. Диагностический характер носит процедура проверки файлов уже находящихся в памяти компьютера. После констатации вирусного заражения наступает этап восстановления «здоровья» вычислительной системы, который может потребовать как весьма жестких мер, когда из системы удаляются все зараженные файлы, так и не столь жестких, когда файлы исправляют, удаляя из них вредоносный код.

Большинство антивирусных программ в той или иной степени расходуют ресурсы тестируемой системы. Иногда это может вызвать заметное снижение скорости выполнения пользовательских приложений. Однако это не должно быть причиной отключения антивирусных проверок, так как ущерб от «работы» вирусов, как правило, с лихвой превышает затраты вычислительных ресурсов и времени пользователя (администратора) на борьбу с вирусами.

Для защиты от вирусов используют три группы методов:

- Методы, основанные на *анализе содержимого файлов* (как файлов данных, так и файлов с кодами команд). К этой группе относятся сканирование сигнатур вирусов, а также проверка целостности и сканирование подозрительных команд.
- Методы, основанные на *отслеживании поведения программ* при их выполнении. Эти методы заключаются в протоколировании всех событий, угрожающих безопасности системы и происходящих либо при реальном выполнении проверяемого кода, либо при его программной эмуляции.
- Методы *регламентации порядка работы* с файлами и программами. Эти методы относятся к административным мерам обеспечения безопасности. Один из наиболее распространенных методов этой группы состоит в том, что в системе (компьютере или корпоративной сети) выполняются только те программы, запись о которых присутствует в списке программ, разрешенных к выполнению в данной системе. Этот список формируется администратором сети из проверенного программного обеспечения.

Сканирование сигнатур

Сигнатура вируса — это уникальная последовательность байтов, которая всегда присутствует в определенном виде вирусов и по которой этот вид вируса можно с большой вероятностью опознать.

Из этого определения следует основная идея метода сканирования сигнатур. Для каждого вновь обнаруженного вируса специалистами выполняется анализ кода, на основании которого определяется сигнатура. Полученный кодовый фрагмент помещают в специальную **базу данных вирусных сигнатур**, с которой работает антивирусная программа.

К размеру сигнатуры предъявляются противоречивые требования. С одной стороны, для того чтобы повысить вероятность правильной диагностики вируса, сигнатура должна быть достаточно длиной — как минимум 8–12 байт, а еще лучше 64 байта. С другой стороны, учитывая огромное число существующих к настоящему моменту вирусов (сотни тысяч), увеличение длины сигнатуры увеличит и без того большой объем базы данных сигнатур.

Система сканирования сигнатур работает следующим образом. Содержимое тестируемого файла сравнивается с каждой из заданных в базе данных этой системы сигнатур. Обна-

ружив совпадение, система автоматически ставит подозрительный файл на *карантин*, то есть блокирует файл от возможного использования. Одним из надежных способов такого блокирования является временное шифрование зараженного файла.

ПРИМЕЧАНИЕ

Различные методы шифрования и упаковки вредоносных программ используют и хакеры. После шифрования или архивирования даже известный вирус становится «невидимым» для обычного сканера сигнатур.

Затем система сканирования оповещает своего пользователя об обнаружении зараженных файлов и о своих действиях, предпринятых по отношению к ним, а также предлагает пользователю выбрать тот или иной вариант дальнейших действий. В частности, она может предложить удалить файл или попытаться восстановить файл путем удаления вредоносного кода и, возможно, реконструкции его исходной структуры.

Процедура сканирования может выполняться как для отдельных файлов, так и для содержимого всего диска, как регулярно, в соответствии с заранее заданным расписанием, так и время от времени по инициативе пользователя. Некоторые антивирусные системы выполняют сканирование файлов синхронно с выполнением тех или иных операций с файлами: открытием, закрытием файлов или отправкой их в виде почтовых вложений; иногда такая тактика помогает быстрее обнаружить появление вируса.

К достоинствам данного метода относят относительно низкую долю ложных срабатываний. Главным же недостатком является *принципиальная невозможность обнаружить присутствие в системе нового вируса*, для которого еще нет сигнатуры в базе данных антивирусной программы. Кроме того, создание базы данных сигнатур является делом очень трудоемким, а ее эксплуатация требует постоянного оперативного обновления, что может представлять проблему как для производителей, так и для пользователей антивирусных средств.

Метод контроля целостности

Метод контроля целостности основывается на том, что любое неожиданное и беспричинное изменение данных на диске является подозрительным событием, требующим особого внимания антивирусной системы. Действительно, любой вирус обязательно оставляет свидетельства своего пребывания на диске. Такими «следами» может быть искажение данных в уже существующих файлах или появление новых исполняемых файлов.

Факт изменения данных — *нарушение целостности* — легко устанавливается путем сравнения контрольной суммы (или дайджеста), заранее подсчитанной для исходного состояния тестируемого кода, и контрольной суммы (дайджеста) текущего состояния тестируемого кода. Если они не совпадают, значит, целостность нарушена и имеются все основания провести для этого кода дополнительную проверку, например, путем сканирование вирусных сигнатур.

В отличие от сканирования сигнатур метод контроля целостности позволяет обнаруживать следы деятельности *любых*, в том числе неизвестных, вирусов, для которых в базе данных еще нет сигнатур. Кроме того, он работает быстрее, поскольку операции подсчета контрольных сумм требуют меньше вычислений, чем операции сравнения кодовых фрагментов.

Сканирование подозрительных команд

В арсенале вирусных программ есть особенно опасные средства. Примером такого грозного оружия может служить код, вызывающий форматирование жесткого диска. Каждый случай обнаружения такого кода должен переводить систему в состояние тревоги, или, по крайней мере, система должна уведомить пользователя об этом событии и попросить подтверждения, прежде чем выполнить операцию, которая может привести к катастрофическим последствиям.

Известно, что вирусные программы разных видов могут содержать функционально подобные (но программно не идентичные) блоки. Например, многие виды вирусов содержат функцию внедрения в исполняемый код. Для этого они сначала отыскивают файлы с расширениями exe, а затем выполняют для них операции открытия и записи. И хотя совокупность этих действий может быть реализована разными кодовыми последовательностями, ее все же можно характеризовать некоторыми общими признаками, которые могут стать опознавательным знаком для функции внедрения вируса.

Если в результате сканирования в файле обнаруживают некоторое число подозрительных команд и/или признаков подозрительных кодовых последовательностей, то делается предположение о вредоносной сущности файла и предпринимаются дополнительные действия по его проверке.

Этот метод обладает хорошим быстродействием, но довольно часто он не способен выявлять новые вирусы.

Отслеживание поведения программ

Принципиально другим подходом по сравнению с методами сканирования содержимого файлов являются методы, основанные на анализе поведения программ *во время их выполнения*. Этот метод обнаружения вирусов можно сравнить с поимкой преступника «за руку» на месте преступления. Тестируемую программу запускают на выполнение, инструкцию за инструкцией, но все ее подозрительные действия контролируются и протоколируются антивирусной системой. Если программа пытается выполнить какую-либо потенциально опасную команду, например записать данные в исполняемый файл другой программы, то ее работа приостанавливается, и антивирусная система запрашивает пользователя о том, какие действия ей надо предпринять.

Антивирусные средства данного типа часто требуют активного участия в тестировании пользователя, призванного реагировать на многочисленные предупреждения системы, значительная часть которых может оказаться впоследствии ложными тревогами.

ПРИМЕЧАНИЕ

Важной характеристикой любого антивирусного средства является частота ложных положительных («Да, это вирус») и ложных отрицательных («Нет, вирус отсутствует») заключений. Если система слишком часто бьет ложную тревогу, то пользователь этой системы может вообще перестать реагировать на эти сигналы, однако если она слишком часто объявляет зараженный файл «чистым», то возникает вопрос о качестве антивирусного средства.

При использовании антивирусных систем, анализирующих поведение программ, всегда существует риск «пропустить удар» от вируса, в результате которого по ошибке будет

выполнена команда вирусного кода, способная нанести ущерб защищаемому компьютеру или сети.

Для устранения этого недостатка был разработан другой метод, который тоже строит работу по распознаванию вирусов на основе анализа выполнения программ, однако тестируемая программа выполняется в искусственно созданной (виртуальной) вычислительной среде, которую иногда называют *песочницей* (sandbox). Такой способ называют **эмуляцией**. При эмуляции так же, как и при реальном выполнении, фиксируются все подозрительные действия программы, однако в этом случае отсутствует риск повреждения информационного окружения.

Принцип работы антивирусных средств, построенных на основе анализа поведения программ, показывает, что эти средства могут использоваться для обнаружения не только известных, но и *не известных* вредоносных программ.

Сетевые экраны

Сетевой, или **межсетевой**, **экран** — это комплекс программно-аппаратных средств, осуществляющий информационную защиту одной части компьютерной сети от другой путем анализа проходящего между ними трафика.

Для сетевых экранов существуют и другие термины, хорошо отражающие функциональное назначение средств защиты этого типа:

- **Брандмауэр** — это слово много лет назад пришло в русский язык из немецкого. Изначально оно обозначало перегородку в поезде, отделяющую область топки паровоза от пассажирского отделения.
- **Файервол** и другие транслитерации английского слова firewall, хотя официально не приняты, можно встретить в литературе достаточно часто. Исходным значением этого термина является элемент конструкции дома, а именно стена, сделанная из огнеупорного материала и препятствующая распространению огня между частями дома (обычно принадлежащими разным собственникам).

Для сетевого экрана одна часть сети является *внутренней*, другая — *внешней* (рис. 24.21). Сетевой экран защищает внутреннюю сеть (например, локальную сеть предприятия или, как вырожденный случай, отдельный компьютер пользователя) от угроз, исходящих из внешней сети (мы будем, как правило, подразумевать под такой сетью Интернет).

Защиту границ между локальными сетями предприятия и Интернетом обеспечивают **корпоративные сетевые экраны**, те же функции, но на границе между домашним компьютером и Интернетом, выполняют **персональные сетевые экраны**.

Для эффективного выполнения сетевым экраном его главной функции — защиты — необходимо, чтобы через него проходил *весь* трафик, которым обмениваются узлы защищаемой части сети с узлами Интернета.

Такое расположение позволяет сетевому экрану полностью контролировать (запрещать, ограничивать или протоколировать) доступ внешних пользователей к ресурсам внутренней

сети. Сетевой экран защищает сеть не только от несанкционированного доступа внешних злоумышленников, но от ошибочных действий пользователей защищаемой сети, например таких, как передача во внешнюю сеть конфиденциальной информации.

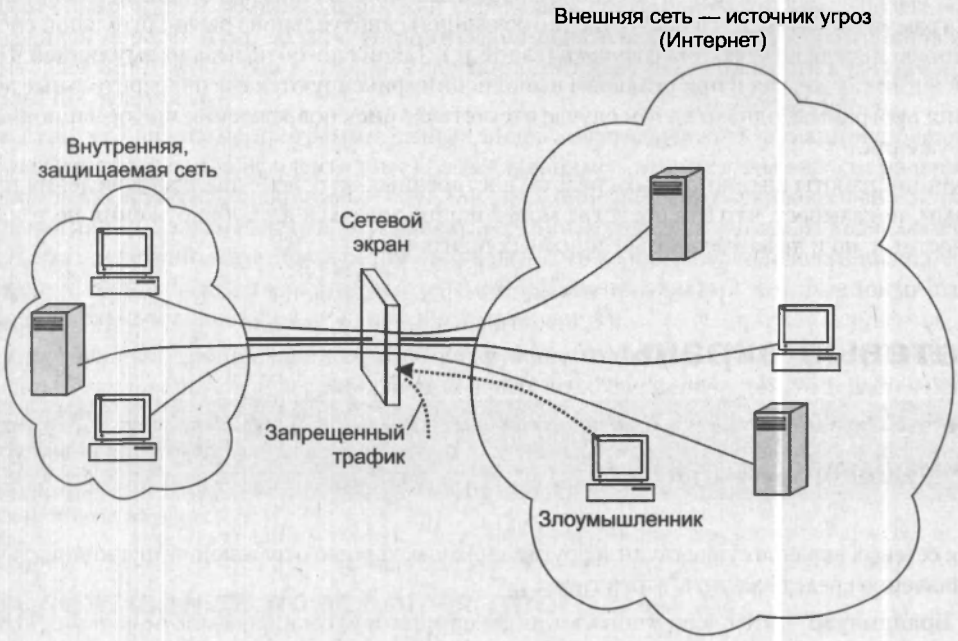


Рис. 24.21. Сетевой экран защищает внутреннюю сеть от угроз, исходящих из внешней сети

Чтобы осуществлять контроль доступа, сетевой экран должен уметь выполнять следующие функции:

- анализировать, контролировать и регулировать трафик (функция фильтрации);
- играть роль логического посредника между внутренними клиентами и внешними серверами (функция прокси-сервера);
- фиксировать все события, связанные с безопасностью (функция аудита).

Наряду с этими базовыми функциями на сетевой экран могут быть возложены и другие вспомогательные функции защиты, в частности:

- антивирусная защита;
- шифрование трафика;
- фильтрация сообщений по содержимому, включая типы передаваемых файлов, имена DNS и ключевые слова;
- предупреждение и обнаружение вторжений и сетевых атак;
- функции VPN;
- трансляция сетевых адресов.

Как можно заметить, большинство из перечисленных функций реализуются в виде отдельных продуктов или в составе систем защиты других типов. Так, функции пакетной

фильтрации встроены практически во все маршрутизаторы, задача обнаружения вирусов решается множеством разнообразных программ, шифрование трафика — неотъемлемый элемент технологий защищенных каналов и т. д., и т. п. Прокси-серверы часто поставляются в виде приложений, более того, они сами часто интегрируют в себе многие функции, свойственные сетевым экранам, такие, например, как аутентификация, трансляция сетевых адресов или фильтрация по содержимому (**контенту**).

Отсюда возникают сложности при определении понятия «сетевой экран». Например, довольно распространено мнение, что сетевой экран — это пограничное устройство, выполняющее пакетную фильтрацию (то есть маршрутизатор), а прокси-сервер — это совершенно отличный от сетевого экрана инструмент защиты. Другие настаивают, что прокси-сервер является непременным и неотъемлемым атрибутом сетевого экрана. Третьи считают, что сетевым экраном может быть названо только такое программное или аппаратное устройство, которое способно отслеживать состояние потока пакетов в рамках соединения. Мы же в этой книге будем придерживаться широко распространенной точки зрения о том, что сетевой экран — это программно-аппаратный комплекс, выполняющий разнообразные функции по защите внутренней сети, набор которых может меняться в зависимости от типа, модели и конкретной конфигурации сетевого экрана.

ПРИМЕР-АНАЛОГИЯ

Функционально сетевой экран можно сравнить с системой безопасности современного аэропорта. Аналогии здесь достаточно очевидные (рис. 24.22) — самолет соответствует защищаемой внутренней сети, а внешняя сеть, из которой приходит потенциально опасный трафик, — внешнему миру, откуда прибывают будущие пассажиры самолета, готовящегося к полету, при этом не все они приезжают с чистыми и ясными намерениями.

В потоке пассажиров, постоянно входящих в здание аэропорта, могут встречаться различные злоумышленники. Наиболее зловещие — террористы — пытаются пронести на борт взрывчатку (в сетевом мире — пакеты, несущие во внутреннюю сеть вирусы, способные «взорвать» серверы и компьютеры пользователей) или оружие для захвата самолета в воздухе (атака по захвату управления удаленным компьютером). Контрабандисты несут с собой незадекларированные ценности (запрещенный контент), а некоторые личности пытаются попасть в самолет по поддельным документам (несанкционированный доступ к внутренним ресурсам сети).

Для того чтобы отфильтровать трафик пассажиров, система безопасности аэропорта пропускает всех пассажиров и их багаж через единственно возможный путь — зону контроля. Также поступают при защите сети, направляя весь входящий трафик через сетевой экран. В зоне контроля аэропорта применяются разнообразные средства проверки пассажиров и их багажа: сличение паспортов с компьютерной базой данных, а лиц пассажиров — с фотографиями в паспортах; просвечивание сумок и чемоданов; проход пассажиров через металлодетекторы, а при первом подозрении — вытряхивание всех вещей; дотошная ручная проверка сумок и прощупывание пассажиров. Между злоумышленниками и службой безопасности постоянно происходит состязание в коварстве, с одной стороны, и находчивости — с другой. Новые трюки вызывают появление новых способов проверки. Например, пронос взрывчатки в подошве ботинка вызвал к жизни не очень приятную обязательную процедуру прохождения металлоискателя в носках, а использование террористами флаконов для маскировки жидких компонентов бомбы лишило пассажиров возможности брать с собой в кабину шампуни и другие любимые жидкости в больших объемах.

Сетевые экраны тоже пытаются использовать все возможные средства и методы для противостояния разнообразным угрозам. С помощью паролей и цифровых сертификатов они проверяют аутентичность внешних узлов, пытающихся установить соединения с внутренними; отслеживают логику обмена пакетами для того, чтобы отразить атаки, основанные на искажении этой логики;

«просвечивают» содержимое электронных писем и загружаемых документов, пытаясь блокировать запрещенный контент; сканируют загружаемые программы, проверяя их на наличие известных вирусов. Так же как и в зоне контроля аэропорта, здесь постоянно идет соревнование между хакерами, все время изобретающими новые методы атак, и разработчиками сетевых экранов, старающихся эти атаки обнаружить и пресечь.

Рис. 24.22. Зона контроля аэропорта как аналогия сетевого экрана

Типы сетевых экранов разных уровней

Одной из принятых классификаций сетевых экранов является разделение их на типы в зависимости от уровня модели OSI, на котором они работают.

Сетевые экраны сетевого уровня, называемые также **экранами с фильтрацией пакетов** (packet filtering firewall), в полном соответствии со своим названием решают задачу фильтрации пакетов по IP-адресам и портам приложений на основании списков доступа (см. раздел «Фильтрация» в главе 18). Фильтрация на основе статических правил, при которой не отслеживаются состояния соединений, называется **простой фильтрацией** (stateless packet inspection). Этому типу сетевых экранов соответствуют маршрутизаторы. Опытный администратор может задать достаточно изощренные правила фильтрации, учитывающие многие требования, касающиеся защиты ресурсов внутренней сети, тем не менее этот тип сетевых экранов уступает по степени защиты другим типам. Преимуществами брандмауэров сетевого уровня являются простота, невысокая стоимость и минимальное влияние на производительность сети.

Сетевые экраны сеансового уровня отслеживают состояние соединений. Они фиксирует подозрительную активность, направленную на сканирование портов и сбор другой информации о сети. *Отслеживание состояний соединений* заключается в том, что сетевой экран проверяет, насколько соответствует последовательность обмена сообщениями контролируемому протоколу. То есть, например, если клиент посылает TCP-сообщение *SYN*, запрашивающее TCP-соединение, сервер должен отвечать TCP-сообщением *ACK SYN*, а не посылать в ответ, например, свой TCP-запрос *SYN*. После того как сетевой экран установил допустимость TCP-соединения, он начинает работать простым передаточным звеном между клиентом и сервером. Для того чтобы контролировать процесс установления соединения, сетевой экран должен фиксировать для себя текущее состояние соединения, то есть *запоминать*, какое последнее сообщение отправил клиент и какое сообщение он ожидает получить. Такой подход, когда пропускаются только те пакеты, которые удовлетворяют логике работы соответствующего протокола, называют **фильтрацией с учетом контекста** (stateful packet inspection). Благодаря такой способности брандмауэры сетевого уровня могут защищать серверы внутренней сети от различных видов атак, использующих уязвимости протоколов, в частности от DoS-атак.

Сетевые экраны прикладного уровня способны интерпретировать, анализировать и контролировать содержимое сообщений, которыми обмениваются приложения. К этому уровню относят прокси-серверы, о которых мы будем говорить подробнее далее. Прокси-сервер перехватывает запросы клиентов к внешним серверам с тем, чтобы потом отправить их от своего имени. Этот тип сетевых экранов обеспечивает самый высокий уровень защиты, хотя и имеет свои недостатки, например требует больших вычислительных затрат. Кроме того, прокси-серверы могут скрывать адрес «доверившегося» ему клиента, что снижает эффективность других средств защиты.

Реализация

Реализация сетевого экрана так же многовариантна, как и его функциональность. В качестве аппаратной составляющей сетевого экрана может выступать маршрутизатор или комбинация маршрутизаторов, компьютер или комбинация компьютеров, комбинация маршрутизаторов и компьютеров, наконец, это может быть специализированное устройство. Таким же разнообразием отличается и программная составляющая сетевого экрана, имеющая гибкую структуру и включающая в себя различные модули, функции которых могут широко варьироваться.

Сложная структура аппаратных и программных средств сетевого экрана, разнообразие настраиваемых параметров, наборы правил, регламентирующих работу фильтров разного уровня, списки паролей и другой информации для проведения аутентификации, списки прав доступа пользователей к внутренним и внешним ресурсам сети — все это требует от администратора значительной дополнительной работы по конфигурированию. Только в случае качественной настройки аппаратуры и программных модулей сетевой экран действительно может стать краеугольным камнем системы защиты сети предприятия. «Умные» сетевые экраны позволяют администратору упростить эту работу, потому что они требуют только задания высокоуровневых правил политики безопасности сети, которые затем автоматически транслируются в низкоуровневые операции по конфигурированию отдельных функциональных подсистем сетевого экрана.

Архитектура

Простейшей архитектурой сети с сетевым экраном является вариант, когда все функции сетевого экрана реализуются *одним* программно-аппаратным устройством, например маршрутизатором или, как показано на рис. 24.23, универсальным компьютером. Такой способ построения защиты логически самый простой, однако он имеет очевидный недостаток, заключающийся в полной зависимости системы защиты от работоспособности одного звена, в данном случае — компьютера-брандмауэра.

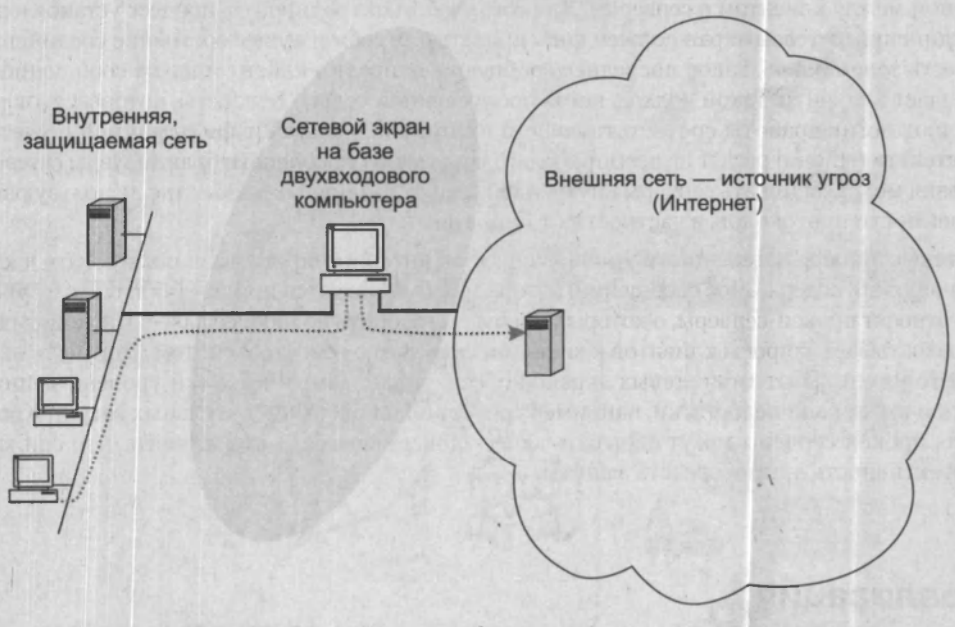


Рис. 24.23. Сетевой экран на базе двухвходового компьютера

Компьютер, играющий роль сетевого экрана, должен иметь, по крайней мере, два сетевых интерфейса, к одному из которых подключается внутренняя, к другому — внешняя сеть. Двухвходовой компьютер выполняет функции программного маршрутизатора, а также те функции сетевого экрана, конкретный перечень которых определяется установленным на данном компьютере программным обеспечением.

Более надежные схемы сетевых экранов включают несколько элементов. В сети, показанной на рис. 24.24, на рубеже защиты установлено два маршрутизатора, между которыми располагается так называемая сеть периметра.

Сеть периметра, или **сеть демилитаризованной зоны** (DMZ), — это сеть, которую для добавления еще одного уровня защиты внутренней сети размещают между внутренней и внешней сетями в качестве буфера.

В сети периметра обычно располагаются компьютеры, которые предоставляют общедоступные сервисы, например почтовый сервер, внешний сервер DNS или внешний веб-сервер предприятия. В этой зоне могут быть размещены также прокси-серверы. Учитывая, что само назначение этих компьютеров предполагает практически никак не ограничиваемый

доступ к ним внешних пользователей (а значит, и злоумышленников), их необходимо защищать особенно тщательно. Главными задачами при защите этих компьютеров (называемых иногда **компьютерами-бастионами**) является обеспечение целостности и доступности размещенных на них данных для пользователей внешней сети. Эту задачу решают «индивидуальные» средства защиты, устанавливаемые на компьютерах-бастионах, такие, например, как антивирусные программы или фильтры спама.

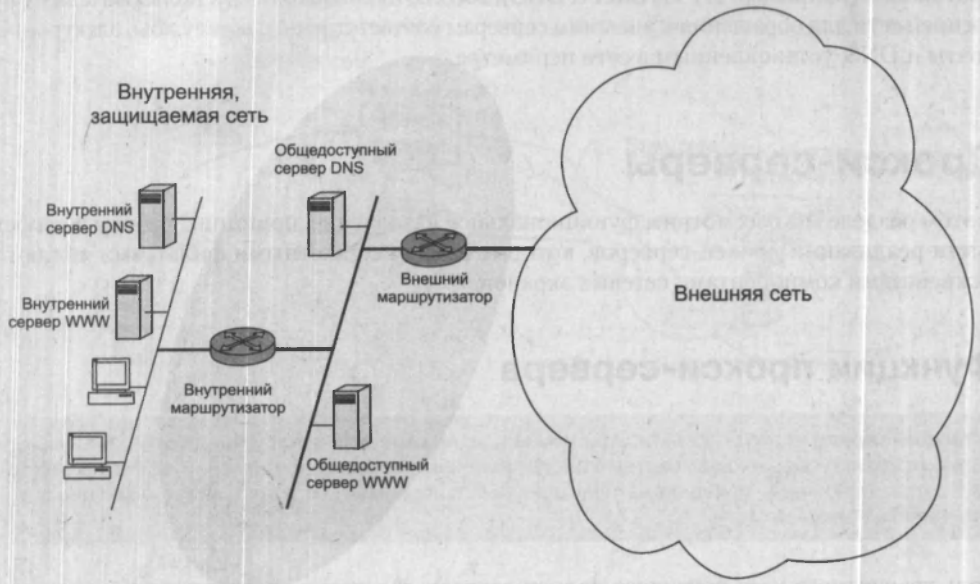


Рис. 24.24. Сетевой экран на базе двух маршрутизаторов

Чтобы пояснить, каким образом сеть периметра усиливает защиту внутренней сети, давайте посмотрим, что произойдет, если какой-либо злоумышленник сможет «взломать» первый рубеж защиты — внешний маршрутизатор — и начнет прослушивать трафик подключенной к нему сети периметра. Очевидно, что он получит доступ только к трафику общедоступных серверов, который не является секретным.

Внешний маршрутизатор призван фильтровать трафик с целью защиты сети периметра и внутренней сети. Однако строгая фильтрация в этом случае оказывается невостребованной. Общедоступные серверы по своей сути предназначены для практически неограниченного доступа. Что касается защиты внутренней сети, правила фильтрации для доступа к ее узлам и сервисам являются одними и теми же для обоих маршрутизаторов, поэтому внешний маршрутизатор может просто положиться в этом деле на внутренний маршрутизатор.

Обычно внешний маршрутизатор находится в зоне ведения *провайдера*, и администраторы корпоративной сети ограничены в возможностях его оперативного реконфигурирования. Это является еще одной причиной, по которой функциональная нагрузка на внешний маршрутизатор обычно невелика.

Основная работа по обеспечению безопасности локальной сети возлагается на *внутренний маршрутизатор*, который защищает ее как от внешней сети, так и от сети периметра. Пра-

вила, определенные для узлов сети периметра по доступу к ресурсам внутренней сети, часто бывают более строгими, чем правила, регламентирующие доступ к этим ресурсам внешних пользователей. Это делается для того, чтобы в случае взлома какого-либо компьютера-бастиона уменьшить число узлов и сервисов, которые впоследствии могут быть атакованы с этого компьютера. Именно поэтому внутренний маршрутизатор должен отбрасывать все пакеты, следующие во внутреннюю сеть из сети периметра, исключая пакеты нескольких протоколов (например, HTTP, SMTP, DNS), абсолютно необходимых пользователям внутренней сети для обращения к внешним серверам соответственно веб-службы, электронной почты и DNS, установленным в сети периметра.

Прокси-серверы

В этом разделе мы рассмотрим функциональное назначение, принципы работы и особенности реализации прокси-серверов, которые наряду с пакетными фильтрами являются важнейшими компонентами сетевых экранов.

Функции прокси-сервера

Прокси-сервер — это особый тип приложения, которое выполняет функции посредника между клиентскими и серверными частями распределенных сетевых приложений, причем предполагается, что клиенты принадлежат внутренней (защищаемой) сети, а серверы — внешней (потенциально опасной) сети.

Роль транзитного узла позволяет прокси-серверу логически разорвать прямое соединение между клиентом и сервером с целью контроля процесса обмена сообщениями между ними.

Подобно сетевому экрану, прокси-сервер может эффективно выполнять свои функции только при условии, что контролируемый им трафик не пойдет обходным путем.

Прокси-сервер может быть установлен не только на платформе, где работают все остальные модули сетевого экрана (рис. 24.25, *а*), но и на любом другом узле внутренней сети или сети периметра (рис. 24.25, *б*). В последнем случае программное обеспечение клиента должно быть сконфигурировано таким образом, чтобы у него не было возможности установить прямое соединение с ресурсным сервером, минуя прокси-сервер.

Когда клиенту необходимо получить ресурс от какого-либо сервера (файл, веб-страницу, почтовое сообщение), он посылает свой запрос прокси-серверу. Прокси-сервер анализирует этот запрос на основании заданных ему администратором правил и решает, каким образом он должен быть обработан (отброшен, передан без изменения ресурсному серверу, модифицирован тем или иным способом перед передачей, немедленно обработан силами самого прокси-сервера).

В качестве правил, которыми руководствуется прокси-сервер, могут выступать условия пакетной фильтрации. Правила могут быть достаточно сложными, например в рабочие часы блокируется доступ к тем или иным узлам и/или приложениям, а доступ к другим узлам разрешается только определенным пользователям, причем для FTP-серверов пользователям разрешается делать лишь загрузку, а выгрузка запрещается. Прокси-серверы

могут также фильтровать почтовые сообщения по типу пересылаемого файла (например, запретить получение сообщений формата MP3) и по их контенту. К разным пользователям могут применяться разные правила фильтрации, поэтому часто на прокси-серверы возлагается задача аутентификации пользователей.

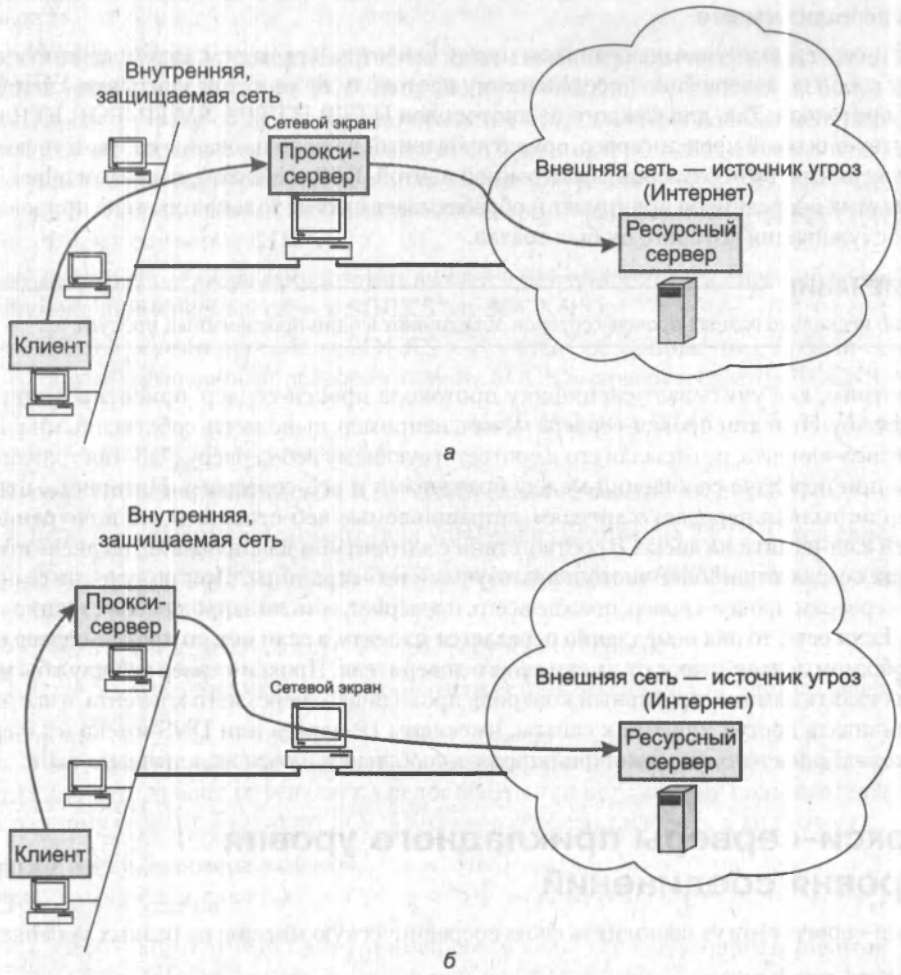


Рис. 24.25. Варианты расположения прокси-серверов: *а* — на сетевом экране, *б* — на узле внутренней сети

Если после всесторонней оценки запроса от приложения прокси-сервер констатирует, что запрос удовлетворяет условиям прохождения дальше во внешнюю сеть, то он выполняет *по поручению приложения, но от своего имени* процедуру соединения с сервером, затребованным данным приложением.

В некоторых случаях прокси-сервер может изменять запрос клиента. Например, если в него встроена функция трансляции сетевых адресов (см. раздел «Трансляция сетевых адресов» в главе 18), он может подменять в пакете запроса IP-адреса и/или номера TCP- и UDP-

портов отправителя. Таким способом прокси-сервер лишает злоумышленника возможности сканировать внутреннюю сеть для получения информации об адресах узлов и структуре сети. Единственный адрес в таком случае, который может узнать злоумышленник, — это адрес компьютера, на котором выполняется программа прокси-сервера. Поэтому многие атаки, построенные на знании злоумышленником адресов узлов внутренней сети, становятся нереализуемыми.

Прокси-сервер, выступая посредником между клиентом и сервером, взаимодействующими между собой по совершенно определенному протоколу, не может не учитывать специфику этого протокола. Так, для каждого из протоколов HTTP, HTTPS, SMTP/POP, FTP, telnet существует особый прокси-сервер, ориентированный на использование соответствующими приложениями: веб-браузером, электронной почтой, FTP-клиентом, клиентом telnet. Каждый из этих посредников принимает и обрабатывает пакеты только того типа приложений, для обслуживания которого он был создан.

ПРИМЕЧАНИЕ

Обычно несколько разных прокси-серверов объединяют в один программный продукт.

Посмотрим, как учитывает специфику протокола прокси-сервер, ориентированный на веб-службу. Этот тип прокси-сервера может, например, выполнить собственными силами запрос веб-клиента, не отсылая его к соответствующему веб-серверу. Работая транзитным узлом при передаче сообщений между браузерами и веб-серверами Интернета, прокси-сервер не только передает клиентам запрашиваемые веб-страницы, но и сохраняет их в своей кэш-памяти на диске. В соответствии с алгоритмом кэширования, на диске прокси-сервера оседают наиболее часто используемые веб-страницы. При получении запросов к веб-серверам прокси-сервер, прежде всего, проверяет, есть ли запрошенная страница в его кэше. Если есть, то она немедленно передается клиенту, а если нет, то прокси-сервер обычным образом делает запрос от имени своего доверителя. Прокси-сервер веб-службы может осуществлять административный контроль проходящего через него контента, в частности ограничивать доступ клиента к сайтам, имеющим IP-адреса или DNS-имена из «черных списков». Более того, он может фильтровать сообщения на основе ключевых слов.

Прокси-серверы прикладного уровня и уровня соединений

Прокси-серверы могут выполнять свою посредническую миссию на разных уровнях.

ПРИМЕР-АНАЛОГИЯ

Рассмотрим пример, иллюстрирующий идею посредничества разного уровня. Для покупки акций инвестор (в нашем случае аналог клиентской части приложения) может прибегнуть к посредническим услугам брокера или трейдера. Брокер, точно следуя указаниям инвестора, покупает для него определенное количество акций определенного типа по определенной цене. Трейдер — это посредник более высокого уровня, которому инвестор поручает самостоятельно принимать решения о необходимых покупках, учитывая различные факторы, например состояние рынка.

Различают прокси-серверы прикладного уровня и уровня соединений.

Прокси-сервер прикладного уровня, как это следует из его названия, умеет «вклиниваться» в процедуру взаимодействия клиента и сервера по одному из прикладных протоколов,

например тому же HTTP, HTTPS, SMTP/POP, FTP или telnet. Чтобы выступать в роли посредника на прикладном уровне, прокси-сервер должен «понимать» смысл команд, «знать» форматы и последовательность сообщений, которыми обмениваются клиент и сервер соответствующей службы. Это дает возможность прокси-серверу проводить анализ содержимого сообщений, делать заключения о подозрительном характере того или иного сеанса.

Прокси-сервер уровня соединений выполняет свою посредническую миссию на транспортном уровне, контролируя TCP-соединение. Очевидно, что работая на более низком уровне, прокси-сервер обладает гораздо меньшим «интеллектом» и имеет меньше возможностей для выявления и предупреждения атак. Однако он обладает одним очень важным преимуществом перед прокси-сервером прикладного уровня — универсальностью, то есть он может быть использован любыми приложениями, работающими по протоколу TCP (а в некоторых случаях и UDP).

Примером прокси-сервера данного типа является разработанный достаточно давно, но все еще широко применяемый сервер **SOCKS** (от SOCKetS).

В простейшей версии протокола SOCKS V4[1] клиент обменивается с прокси-сервером SOCKS двумя сообщениями: запросом клиента SOCKS-серверу и ответом SOCKS-сервера клиенту.

- Запрос клиента SOCKS-серверу:
  - поле 1 — номер версии SOCKS, 1 байт (для этой версии — 4);
  - поле 2 — код команды, 1 байт (для установки соединения TCP/IP код равен 1);
  - поле 3 — номер порта, 2 байта (TCP-порт запрашиваемого пользователем ресурсного сервера, например, для 21 для FTP);
  - поле 4 — IP-адрес, 4 байта (IP-адрес ресурсного сервера);
  - поле 5 — идентификатор пользователя (строка переменной длины, завершаемая байтом null).

  SOCKS-сервер анализирует все полученные данные и на основании сконфигурированных для него правил определяет, предоставить или нет данному пользователю доступ к данному серверу. Результат SOCKS-сервер сообщает клиенту в виде ответа.
- Ответ SOCKS-сервера клиенту:
  - поле 1 — байт null;
  - поле 2 — код ответа, 1 байт (применяются коды для следующих вариантов ответа: запрос разрешен, запрос отклонен или ошибочен, запрос не удался из-за проблем с идентификацией пользователя);
  - несколько байтов, игнорируемых клиентом.

Если прокси-сервер сообщил в ответе, что запрос разрешен, то SOCKS-сервер начинает работать промежуточным звеном между клиентом и сервером (например, FTP), контролируя поток квитанций, которыми они обмениваются.

---

[1] Более поздняя версия протокола SOCKS V5 расширяет возможности версии SOCKS V4, добавляя поддержку UDP, доменных имен, адресов IPv6 и процедур аутентификации пользователей.

«Проксификация» приложений

Заметим, что не каждое приложение, построенное в архитектуре клиент-сервер, непременно *должно* работать через прокси-сервер, а также не каждое из них *имеет возможность* работать через прокси-сервер.

Список приложений (точнее их клиентских частей), которые должны передавать свои запросы во внешнюю сеть исключительно через прокси-сервер, определяется администратором. А чтобы эти приложения имели возможности для такого режима выполнения, их программы должны быть соответствующим образом написаны.

Точнее приложения должны быть оснащены средствами, которые распознавали бы запросы к внешним серверам и перед отправкой преобразовывали эти запросы так, чтобы все они попадали на соответствующий прокси-сервер, а не передавались в соответствии со стандартным протоколом прямо на сервер-адресат. Эти средства должны также поддерживать протокол обмена сообщениями приложения-клиента с прокси-сервером. В последние годы в большинстве приложений, ориентированных на работу через Интернет, предусмотрена *встроенная поддержка прокси-сервера*. Такой поддержкой, например, оснащены все веб-браузеры и все клиенты электронной почты, которыми мы сейчас пользуемся.

«Проксификация» приложения, изначально не рассчитанного на работу через прокси-сервер, требует изменения исходного кода с последующей перекомпиляцией — очевидно, что такая работа не представляет сложностей для разработчиков данного приложения, но не всегда под силу обслуживающему персоналу сети. Задача последних заключается в приобретении готовых приложений, совместимых с используемым в сети прокси-сервером. Однако даже приобретение готового «проксифицированного» клиента не делает его готовым к работе — необходимо еще конфигурирование, в частности нужно сообщить клиенту адрес узла сети, на котором установлен соответствующий прокси-сервер.

Как можно было бы предположить, процедура «проксификации» значительно упрощается для прокси-сервера уровня соединений, в частности SOCKS-сервера. Для «проксификации» приложения в этом случае достаточно внести простейшие исправления в исходный текст, а затем выполнить его перекомпиляцию и связывание с библиотекой процедур SOCKS. Исправления сводятся к замене всех стандартных вызовов сетевых функций версиями этих функций из библиотеки SOCKS, в частности стандартный вызов `listen()` заменяется вызовом `rlisten()`, вызов `bind()` — вызовом `rbind()`, вызов `accept()` — вызовом `raccept()`.

Имеется еще один подход к «проксификации» — встраивание поддержки прокси-сервера в операционную систему. В этом случае приложения могут оставаться в полном «неведении» о существовании в сети прокси-сервера, за них все необходимые действия выполнит ОС.

Помимо основных функций, многие прокси-серверы способны обнаруживать вирусы еще до того, как они попали во внутреннюю сеть. К другим полезным (для администрации и службы безопасности) вспомогательным функциям прокси-сервера относится сбор статистических данных о доступе пользователей в Интернет: когда и какие сайты посещал тот или иной пользователь, сколько времени продолжалось каждое посещение.

Системы обнаружения вторжений

Система обнаружения вторжений (Intrusion Detection System, IDS) — это программное или аппаратное средство, предназначенное для предупреждения, выявления и протоколирования некоторых типов сетевых атак.

В отличие от сетевых экранов и прокси-серверов, которые строят защиту сети исключительно на основе анализа сетевого трафика, системы обнаружения вторжений учитывают в своей работе различные подозрительные события, происходящие в системе.

Существуют ситуации, когда сетевой экран оказывается проницаемым для злоумышленника, например, когда атака идет через туннель VPN из взломанной сети или инициатором атаки является пользователь внутренней сети и т. п. И дело здесь не в плохой конфигурации межсетевого экрана, а в самом принципе его работы. Экран, несмотря на то что обладает памятью и анализирует последовательность событий, конфигурируется на блокирование трафика с заранее предсказуемыми признаками, например по IP-адресам или протоколам. Так что факт взлома внешней сети, с которой у него был установлен защищенный канал и которая до сих пор вела себя вполне корректно, в правилах экрана отразить нельзя. Точно так же, как и неожиданную попытку легального внутреннего пользователя скопировать файл с паролями или повысить уровень своих привилегий. Подобные подозрительные действия может обнаружить только система со встроенными агентами во многих точках сети, причем она должна следить не только за трафиком, но и за обращениями к критически важным ресурсам операционных систем отдельных компьютеров, а также иметь информацию о перечне подозрительных действий (сигнатур атак) пользователей. Таковой и является система обнаружения вторжений. Она не дублирует действия межсетевого экрана, а дополняет их, производя, кроме того, автоматический анализ всех журналов событий, имеющихся у сетевых устройств и средств защиты, чтобы попытаться найти следы атаки, если ее не удалось зафиксировать в реальном времени.

Протоколы защищенного канала. IPsec

Известно, что задачу защиты данных можно разделить на две подзадачи: защиту данных внутри компьютера и защиту данных в процессе их передачи от одного компьютера в другой. Для обеспечения безопасности данных при их передаче по публичным сетям используются различные технологии защищенного канала.

Технология защищенного канала обеспечивает защиту трафика между двумя точками в открытой транспортной сети, например в Интернете. Защищенный канал подразумевает выполнение трех основных функций:

- взаимная аутентификация абонентов при установлении соединения, которая может быть выполнена, например, путем обмена паролями;
- защита передаваемых по каналу сообщений от несанкционированного доступа, например, путем шифрования;
- подтверждение целостности поступающих по каналу сообщений, например, путем передачи одновременно с сообщением его дайджеста.

В зависимости от местарасположения программного обеспечения защищенного канала различают две схемы его образования:

- схема с конечными узлами, взаимодействующими через публичную сеть (рис. 24.26, *а*);
- схема с оборудованием поставщика услуг публичной сети, расположенным на границе между частной и публичной сетями (рис. 24.26, *б*).

Рис. 24.26. Два подхода к образованию защищенного канала

В первом случае защищенный канал образуется программными средствами, установленными на двух удаленных компьютерах, принадлежащих двум разным локальным сетям одного предприятия и связанных между собой через публичную сеть. Преимуществом этого подхода является полная защищенность канала вдоль всего пути следования, а также возможность использования любых протоколов создания защищенных каналов, лишь бы на конечных точках канала поддерживался один и тот же протокол. Недостатки заключаются в избыточности и децентрализованности решения. Избыточность состоит в том, что вряд ли стоит создавать защищенный канал на всем пути следования данных: уязвимыми для злоумышленников обычно являются сети с коммутацией пакетов, а не каналы телефонной сети или выделенные каналы, через которые локальные сети подключены к территориальной сети. Поэтому защиту каналов доступа к публичной сети можно считать избыточной. Децентрализация заключается в том, что для каждого компьютера, которому требуется предоставить услуги защищенного канала, необходимо отдельно устанавливать, конфигурировать и администрировать программные средства защиты данных. Подключение каждого нового компьютера к защищенному каналу требует выполнять эти трудоемкие операции заново.

Во втором случае клиенты и серверы не участвуют в создании защищенного канала — он прокладывается только внутри публичной сети с коммутацией пакетов, например внутри Интернета. Так, канал может быть проложен между сервером удаленного доступа поставщика услуг публичной сети и пограничным маршрутизатором корпоративной сети. Это хорошо масштабируемое решение, управляемое централизовано администраторами

как корпоративной сети, так и сети поставщика услуг. Для компьютеров корпоративной сети канал прозрачен — программное обеспечение этих конечных узлов остается без изменений. Такой гибкий подход позволяет легко образовывать новые каналы защищенного взаимодействия между компьютерами независимо от места их расположения. Реализация этого подхода сложнее — нужен стандартный протокол образования защищенного канала, требуется установка у всех поставщиков услуг программного обеспечения, поддерживающего такой протокол, необходима поддержка протокола производителями пограничного коммуникационного оборудования. Однако вариант, когда все заботы по поддержанию защищенного канала берет на себя поставщик услуг публичной сети, оставляет сомнения в надежности защиты: во-первых, незащищенными оказываются каналы доступа к публичной сети, во-вторых, потребитель услуг чувствует себя в полной зависимости от надежности поставщика услуг.

Иерархия технологий защищенного канала

Защищенный канал можно построить с помощью системных средств, реализованных на разных уровнях модели OSI (рис. 24.27).

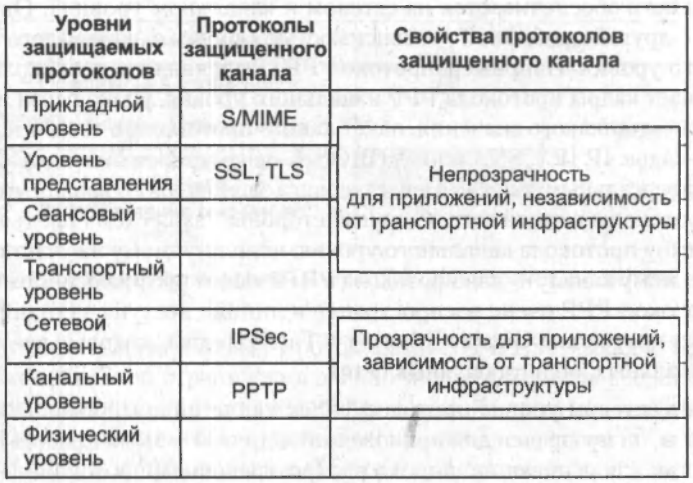


Рис. 24.27. Протоколы, формирующие защищенный канал на разных уровнях модели OSI

Если защита данных осуществляется средствами верхних уровней (прикладного, представления или сеансового), то такой способ защиты не зависит от технологий транспортировки данных (IP или IPX, Ethernet или ATM), что можно считать несомненным достоинством. В то же время приложения при этом становятся зависимыми от конкретного протокола защищенного канала, так как в них должны быть встроены явные вызовы функций этого протокола.

Защищенный канал, реализованный на самом высоком (прикладном) уровне, защищает только вполне определенную сетевую службу, например файловую, гипертекстовую или почтовую. Так, протокол S/MIME защищает исключительно сообщения электронной почты. При таком подходе для каждой службы необходимо разрабатывать собственную защищенную версию протокола.

Популярный протокол **SSL**[1] (Secure Socket Layer — слой защищенных сокетов) работает на уровне представления и создает защищенный канал, используя следующие технологии безопасности:

- взаимная аутентификация приложений на обоих концах защищенного канала выполняется путем обмена сертификатами (стандарт X.509);
- для контроля целостности передаваемых данных используются дайджесты;
- секретность обеспечивается шифрацией со средствами симметричных ключей сеанса.

Протокол SSL разработан компанией Netscape Communications для защиты данных, передаваемых между веб-сервером и веб-браузером, но он может быть использован и любыми другими приложениями. Работа протокола защищенного канала на уровне представления делает его более универсальным средством, чем протокол безопасности прикладного уровня. Однако для того чтобы приложение смогло воспользоваться протоколом уровня представления, в него по-прежнему приходится вносить исправления, хотя и не столь существенные, как в случае протокола прикладного уровня. Модификация приложения в данном случае сводится к встраиванию явных обращений к API соответствующего протокола безопасности.

Средства защищенного канала становятся прозрачными для приложений в тех случаях, когда безопасность обеспечивается на сетевом и канальном уровнях. Однако здесь мы сталкиваемся с другой проблемой — зависимостью сервиса защищенного канала от протокола нижнего уровня. Например, протокол PPTP, не являясь протоколом канального уровня, защищает кадры протокола PPP канального уровня, упаковывая их в IP-пакеты. При этом не имеет никакого значения, пакет какого протокола, в свою очередь, упакован в данном PPP-кадре: IP, IPX, SNA или NetBIOS. С одной стороны, это делает сервис PPTP достаточно универсальным, так как клиент сервиса защищенного канала может задействовать любые протоколы в своей сети. С другой стороны, такая схема предъявляет жесткие требования к типу протокола канального уровня, используемому на участке доступа клиента к защищенному каналу — для протокола PPTP таким протоколом может быть *только PPP*. Хотя протокол PPP очень распространен в линиях доступа, сегодня конкуренцию ему составляют протоколы Gigabit Ethernet и Fast Ethernet, которые все чаще работают не только в локальных, но и глобальных сетях.

Работающий на сетевом уровне протокол IPSec является компромиссным вариантом. С одной стороны, он прозрачен для приложений, с другой — может работать практически во всех сетях, так как основан на широко распространенном протоколе IP и использует любую технологию канального уровня (PPP, Ethernet, ATM и т. д.).

Распределение функций между протоколами IPSec

Протокол IPSec называют в стандартах Интернета *системой*. Действительно, IPSec — это согласованный набор открытых стандартов, имеющий сегодня вполне очерченное ядро, которое в то же время может быть достаточно просто дополнено новыми функциями и протоколами.

[1] 95 % веб-сайтов в Великобритании, принимающих от клиентов информацию о кредитных и дебетовых карточках, используют для передачи такого рода данных протокол SSL.

Ядро IPSec составляют три протокола:

- AH (Authentication Header — заголовок аутентификации) — гарантирует целостность и аутентичность данных;
- ESP (Encapsulating Security Payload — инкапсуляция зашифрованных данных) — шифрует передаваемые данные, обеспечивая конфиденциальность, может также поддерживать аутентификацию и целостность данных;
- IKE (Internet Key Exchange — обмен ключами Интернета) — решает вспомогательную задачу автоматического предоставления конечным точкам защищенного канала секретных ключей, необходимых для работы протоколов аутентификации и шифрования данных.

Как видно из краткого описания функций, возможности протоколов AH и ESP частично перекрываются (рис. 24.28). В то время как AH отвечает только за обеспечение целостности и аутентификации данных, ESP может шифровать данные и, кроме того, выполнять функции протокола AH (хотя, как увидим позднее, аутентификация и целостность обеспечиваются им в несколько урезанном виде). ESP может поддерживать функции шифрования и аутентификации/целостности в любых комбинациях, то есть либо всю группу функций, либо только аутентификацию/целостность, либо только шифрование.

<table>
<tr><th>Выполняемые функции</th><th colspan="2">Протокол</th></tr>
<tr><td>Обеспечение целостности</td><td rowspan="2">AH</td><td rowspan="3">ESP</td></tr>
<tr><td>Обеспечение аутентичности</td></tr>
<tr><td>Обеспечение конфиденциальности (шифрование)</td><td></td></tr>
<tr><td>Распределение секретных ключей</td><td colspan="2">IKE</td></tr>
</table>

Рис. 24.28. Распределение функций между протоколами IPSec

Разделение функций защиты между протоколами AH и ESP вызвано применяемой во многих странах практикой ограничения экспорта и/или импорта средств, обеспечивающих конфиденциальность данных путем шифрования. Каждый из этих протоколов может использоваться как самостоятельно, так и одновременно с другим, так что в тех случаях, когда шифрование из-за действующих ограничений применять нельзя, систему можно поставлять только с протоколом AH. Естественно, подобная защита данных во многих случаях оказывается недостаточной. Принимающая сторона получает лишь возможность проверить, что данные были отправлены именно тем узлом, от которого они ожидаются, и дошли в том виде, в котором были отправлены. Однако от несанкционированного просмотра данных на пути их следования по сети протокол AH защитить не может, так как не шифрует их. Для шифрования данных необходим протокол ESP.

Безопасная ассоциация

Для того чтобы протоколы AH и ESP могли выполнять свою работу по защите передаваемых данных, протокол IKE устанавливает между двумя конечными точками логическое соединение (рис. 24.29), которое в стандартах IPSec носит название **безопасной ассоциации** (Security Association, SA).

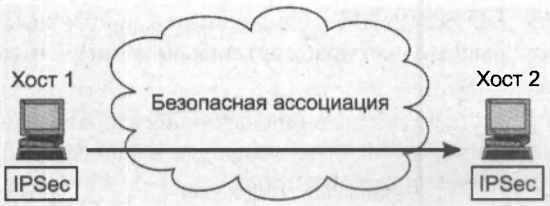


Рис. 24.29. Безопасная ассоциация

Стандарты IPSec позволяют конечным точкам защищенного канала использовать как одну безопасную ассоциацию для передачи трафика всех взаимодействующих через этот канал хостов, так и создавать для этой цели произвольное число безопасных ассоциаций, например, по одной на каждое TCP-соединение. Это дает возможность выбирать нужную степень детализации защиты — от одной общей ассоциации для трафика множества конечных узлов до индивидуально настроенных ассоциаций для защиты каждого приложения.

Безопасная ассоциация в протоколе IPSec представляет собой однонаправленное (симплексное) логическое соединение, поэтому если требуется обеспечить безопасный двусторонний обмен данными, необходимо установить две безопасные ассоциации. Эти ассоциации в общем случае могут иметь разные характеристики, например, в одну сторону при передаче запросов к базе данных достаточно только аутентификации, а для ответных данных, несущих ценную информацию, дополнительно нужно обеспечить конфиденциальность.

Установление безопасной ассоциации начинается с взаимной аутентификации сторон, потому что все меры безопасности теряют смысл, если данные передаются или принимаются не тем лицом или не от того лица. Выбираемые далее параметры SA определяют, какой из двух протоколов, AH или ESP, будет применяться для защиты данных, какие функции будет выполнять протокол (например, можно выполнять только аутентификацию и проверку целостности или, кроме того, еще и обеспечивать конфиденциальность). Очень важными параметрами безопасной ассоциации являются также секретные ключи, используемые в работе протоколов AH и ESP.

Протокол IPSec допускает как автоматическое, так и ручное установление безопасной ассоциации. При ручном способе администратор конфигурирует конечные узлы так, чтобы они поддерживали согласованные параметры ассоциации, включая секретные ключи. При автоматической процедуре установления SA протоколы IKE, работающие по разные стороны канала, выбирают параметры в ходе переговорного процесса. Для каждой задачи, решаемой протоколами AH и ESP, предлагается несколько схем аутентификации и шифрования (рис. 24.30). Это делает протокол IPSec очень гибким средством. Заметим, что выбор дайджест-функции для решения задач целостности и аутентификации никак не влияет на выбор функции шифрования, обеспечивающей конфиденциальность данных.

Для обеспечения совместимости в стандартной версии IPsec определен некоторый обязательный «инструментальный» набор, в частности для аутентификации данных всегда может быть использована одна из стандартных дайджест-функций MD5 либо SHA-1, а в число алгоритмов шифрования непременно входит DES. При этом производители продуктов, в которых используется IPSec, вольны расширять протокол путем включения других алгоритмов аутентификации и симметричного шифрования, что они с успехом и де-

лают. Например, многие реализации IPSec поддерживают популярный алгоритм шифрования Triple DES, а также сравнительно новые алгоритмы: Blowfish, Cast, CDMF, Idea, RC5.

Рис. 24.30. Согласование параметров в протоколе ESP

Транспортный и туннельный режимы

Протоколы AH и ESP могут защищать данные в двух режимах: транспортном и туннельном. В **транспортном режиме** передача IP-пакета через сеть выполняется с помощью оригинального заголовка этого пакета, а в **туннельном режиме** исходный пакет помещается в новый IP-пакет, и передача данных по сети выполняется на основании заголовка нового IP-пакета.

Применение того или иного режима зависит от требований, предъявляемых к защите данных, а также от роли, которую играет в сети узел, завершающий защищенный канал. Так, узел может быть хостом (конечным узлом) или шлюзом (промежуточным узлом). Соответственно, имеются три схемы применения протокола IPSec:

- хост-хост;
- шлюз-шлюз;
- хост-шлюз.

В схеме хост-хост защищенный канал, или, что в данном контексте одно и то же, безопасная ассоциация, устанавливается между двумя конечными узлами сети (см. рис. 24.29). Тогда протокол IPSec работает на конечных узлах и защищает данные, передаваемые от хоста 1 к хосту 2. Для схемы хост-хост чаще всего используется транспортный режим защиты.

В соответствии со схемой шлюз-шлюз защищенный канал устанавливается между двумя промежуточными узлами, так называемыми **шлюзами безопасности** (Security Gateway, SG), на каждом из которых работает протокол IPSec (рис. 24.31). Защищенный обмен данными может происходить между любыми двумя конечными узлами, подключенными к сетям, которые расположены позади шлюзов безопасности. От конечных узлов поддержка протокола IPSec не требуется, они передают свой трафик в незащищенном виде через заслуживающие доверие внутренние сети предприятий. Трафик, направляемый в общедоступную сеть, проходит через шлюз безопасности, который и обеспечивает его защиту с помощью протокола IPSec. Шлюзам доступен только туннельный режим работы.

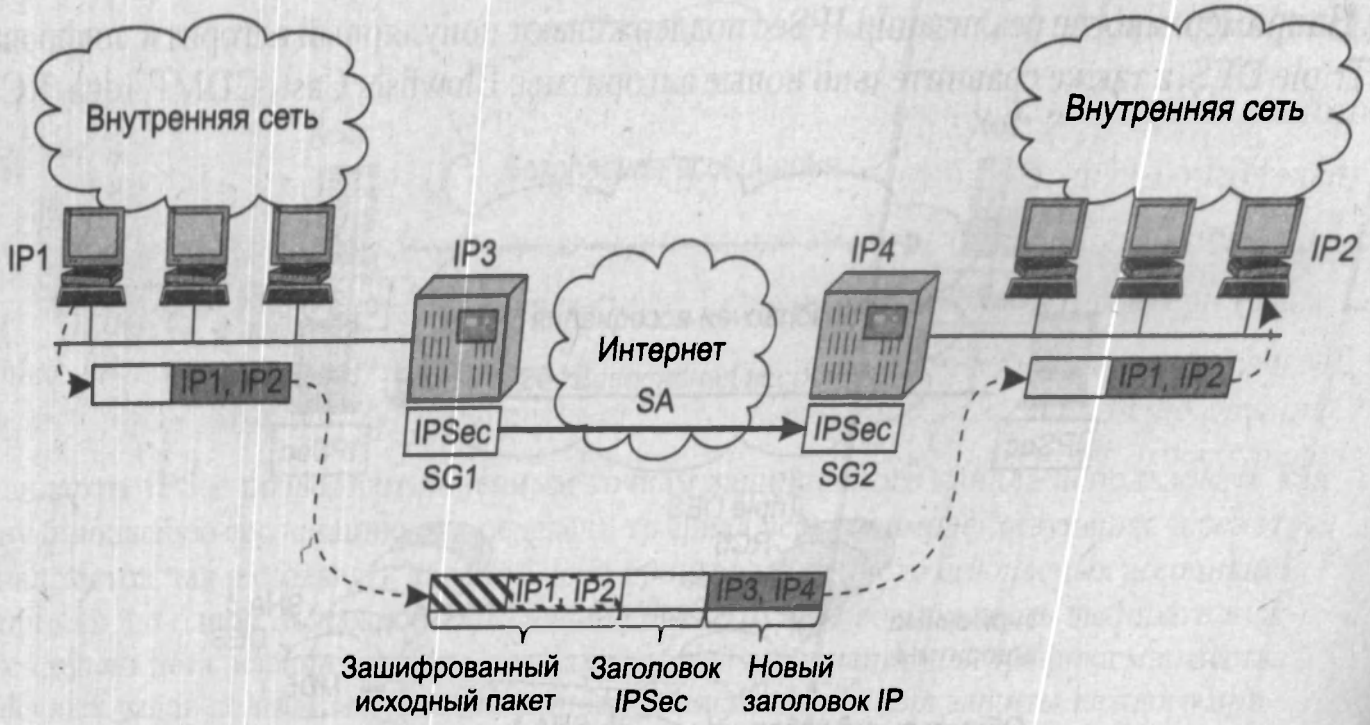


Рис. 24.31. Работа защищенного канала по схеме шлюз-шлюз в туннельном режиме

На рис. 24.31 пользователь компьютера с адресом IP1 посылает пакет по адресу IP2, используя туннельный режим протокола IPSec. Шлюз SG1 зашифровывает пакет целиком, вместе с заголовком, и снабжает его новым заголовком IP, в котором в качестве адреса отправителя указывает свой адрес — IP3, а в качестве адреса получателя — адрес IP4 шлюза SG2. Вся передача данных по составной IP-сети выполняется на основании заголовка внешнего пакета, а внутренний пакет становится при этом полем данных для внешнего пакета. На шлюзе SG2 протокол IPSec извлекает инкапсулированный пакет и расшифровывает его, приводя к исходному виду.

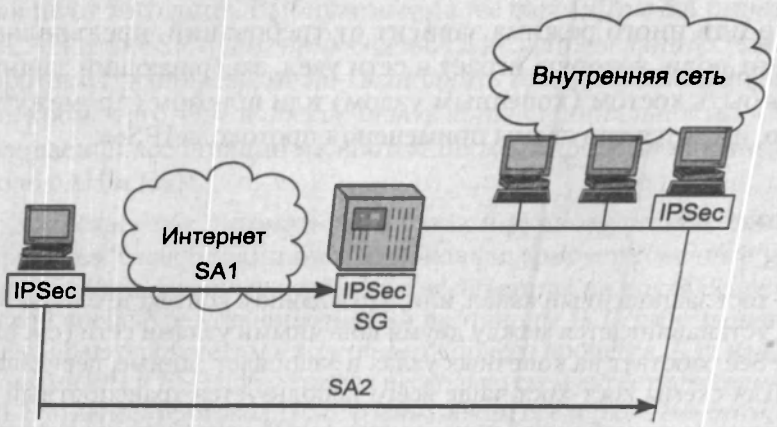


Рис. 24.32. Схема защищенного канала хост-шлюз

Схема хост-шлюз часто применяется при удаленном доступе. В этом случае защищенный канал прокладывается между удаленным хостом, на котором работает протокол IPSec, и шлюзом, защищающим трафик для всех хостов, входящих во внутреннюю сеть предприятия. Эту схему можно усложнить, создав параллельно еще один защищенный канал — между удаленным хостом и каким-либо хостом, принадлежащим внутренней сети, защищаемой шлюзом (рис. 24.32). Такое комбинированное использование двух безопасных ассоциаций позволяет надежно защитить трафик и во внутренней сети.

Протокол AH

Протокол AH позволяет приемной стороне убедиться, что:

- пакет был отправлен стороной, с которой установлена безопасная ассоциация;
- содержимое пакета не было искажено в процессе его передачи по сети;
- пакет не является дубликатом уже полученного пакета.

Две первые функции обязательны для протокола AH, а последняя выбирается при установлении ассоциации по желанию. Для выполнения этих функций протокол AH использует специальный заголовок (рис. 24.33).

Рис. 24.33. Структура заголовка протокола AH

В поле *следующего заголовка* (next header) указывается код протокола более высокого уровня, то есть протокола, сообщение которого размещено в поле данных IP-пакета. Скорее всего, им будет один из протоколов транспортного уровня (TCP или UDP) или протокол ICMP, но может встретиться и протокол ESP, если он используется в комбинации с AH.

В поле *длины полезной нагрузки* (payload length) содержится длина заголовка AH.

Индекс параметров безопасности (Security Parameters Index, SPI) служит для связи пакета с предусмотренной для него безопасной ассоциацией. Немного позже мы обсудим его более подробно.

Поле *порядкового номера* (Sequence Number, SN) указывает на порядковый номер пакета и применяется для защиты от его ложного воспроизведения (когда третья сторона пытается повторно использовать перехваченные защищенные пакеты, отправленные реально аутентифицированным отправителем). Отправляющая сторона последовательно увеличивает значение этого поля в каждом новом пакете, передаваемом в рамках данной ассоциации, так что приход дубликата обнаружится принимающей стороной (если, конечно, в рамках ассоциации будет активирована функция защиты от ложного воспроизведения). Однако в любом случае в функции протокола AH не входит восстановление утерянных и упорядочивание прибывающих пакетов — он просто отбрасывает пакет, когда обнаруживает, что аналогичный пакет уже получен. Чтобы сократить требуемую для работы протокола буферную память, используется механизм скользящего окна — на предмет дублирования проверяются только те пакеты, чей номер находится в пределах окна. Окно обычно выбирается размером в 32 или 64 пакета.

Поле *данных аутентификации* (authentication data), которое содержит так называемое **значение проверки целостности** (Integrity Check Value, ICV), служит для аутентификации и проверки целостности пакета. Это значение является дайджестом, вычисляемым с помощью одной из двух обязательно поддерживаемых протоколом AH односторонних функций шифрования MD5 или SAH-1, но может использоваться и любая другая функ-

ция, о которой стороны договорились в ходе установления ассоциации. При вычислении дайджеста пакета в качестве параметра ОФШ выступает симметричный секретный ключ, который был задан для данной ассоциации вручную или автоматически с помощью протокола IKE. Так как длина дайджеста зависит от выбранной ОФШ, это поле имеет в общем случае переменный размер.

Протокол AH старается охватить при вычислении дайджеста как можно большее число полей исходного IP-пакета, но некоторые из них в процессе передачи пакета по сети меняются непредсказуемым образом, поэтому не могут быть включены в аутентифицируемую часть пакета. Например, целостность значения поля времени жизни (TTL) в приемной точке канала оценить нельзя, так как оно уменьшается на единицу каждым промежуточным маршрутизатором и никак не может совпадать с исходным.

Местоположение заголовка AH в пакете зависит от того, в каком режиме — транспортном или туннельном — сконфигурирован защищенный канал. Результирующий пакет в транспортном режиме выглядит так, как показано на рис. 24.34.

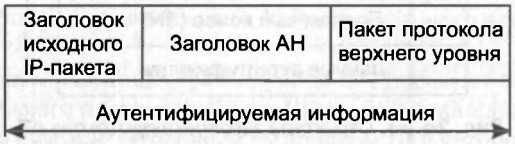


Рис. 24.34. Структура IP-пакета, обработанного протоколом AH в транспортном режиме

При использовании туннельного режима, когда шлюз IPSec принимает проходящий через него транзитом исходящий пакет и создает для него внешний IP-пакет, протокол AH защищает все поля исходного пакета, а также неизменяемые поля нового заголовка внешнего пакета (рис. 24.35).

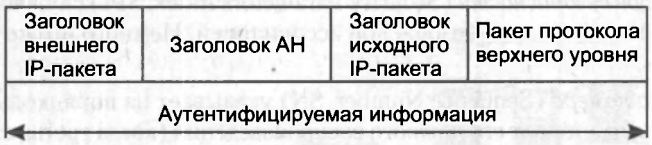


Рис. 24.35. Структура IP-пакета, обработанного протоколом AH в туннельном режиме

Протокол ESP

Протокол ESP решает две группы задач. К первой относятся задачи обеспечения аутентификации и целостности данных на основе дайджеста, аналогичные задачам протокола AH, ко второй — защита передаваемых данных путем их шифрования от несанкционированного просмотра.

Как видно на рис. 24.36, заголовок ESP делится на две части, разделяемые полем данных. Первая часть, называемая собственно *заголовком ESP*, образуется двумя полями (SPI и SN), назначение которых аналогично одноименным полям протокола AH, и размещается перед полем данных. Остальные служебные поля протокола ESP, называемые *концевиком ESP*, расположены в конце пакета.

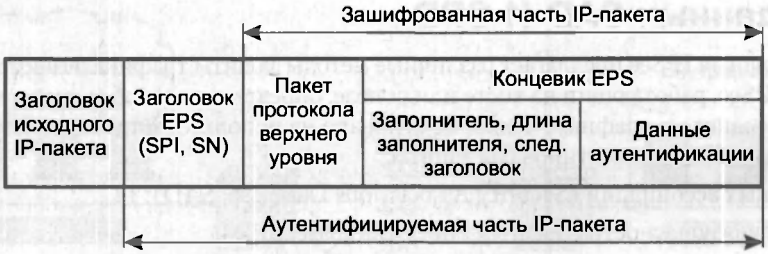


Рис. 24.36. Структура IP-пакета, обработанного протоколом ESP в транспортном режиме

Два поля концевика — *следующего заголовка* и *данных аутентификации* — также аналогичны полям заголовка AH. Поле данных аутентификации отсутствует, если при установлении безопасной ассоциации принято решение не использовать возможностей протокола ESP, касающихся обеспечения целостности. Помимо этих полей концевик содержит два дополнительных поля — *заполнителя* и *длины заполнителя*. Заполнитель может понадобиться в трех случаях. Во-первых, для нормальной работы некоторых алгоритмов шифрования необходимо, чтобы шифруемый текст содержал кратное число блоков определенного размера. Во-вторых, формат заголовка ESP требует, чтобы поле данных заканчивалось на границе четырех байтов. И наконец, заполнитель можно использовать, чтобы скрыть действительный размер пакета в целях обеспечения так называемой частичной конфиденциальности трафика. Правда, возможность маскировки ограничивается сравнительно небольшим объемом заполнителя — 255 байт, поскольку большой объем избыточных данных может снизить полезную пропускную способность канала связи.

На рис. 24.36 показано размещение полей заголовка ESP в *транспортном режиме*. В этом режиме ESP не шифрует заголовок IP-пакета, иначе маршрутизатор не сможет прочитать поля заголовка и корректно осуществить продвижение пакета между сетями. В число шифруемых полей не попадают также поля SPI и SN, которые должны передаваться в открытом виде для того, чтобы прибывший пакет можно было отнести к определенной ассоциации и предотвратить ложное воспроизведение пакета.

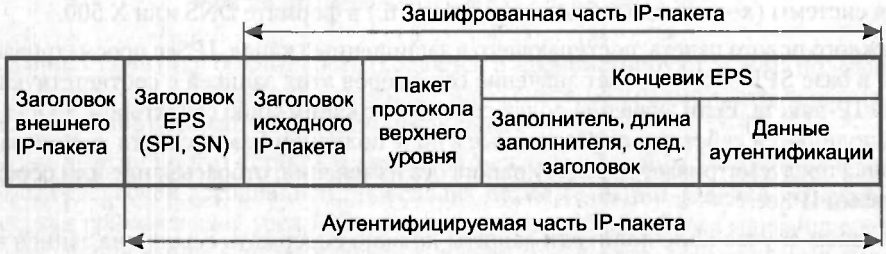


Рис. 24.37. Структура IP-пакета, обработанного протоколом ESP в туннельном режиме

В *туннельном режиме* заголовок исходного IP-пакета помещается после заголовка ESP и полностью попадает в число защищаемых полей, а заголовок внешнего IP-пакета протоколом ESP не защищается (рис. 24.37).

Базы данных SAD И SPD

Итак, технология IPSec предлагает различные методы защиты трафика. Каким же образом протокол IPSec, работающий на хосте или шлюзе, определяет способ защиты, который он должен применить к трафику? Решение основано на использовании в каждом узле, поддерживающем IPSec, двух типов баз данных:

- безопасных ассоциаций (Security Associations Database, SAD);
- политики безопасности (Security Policy Database, SPD).

При установлении безопасной ассоциации, как и при любом другом логическом соединении, две стороны принимают ряд соглашений, регламентирующих процесс передачи потока данных между ними. Соглашения фиксируются в виде набора параметров. Для безопасной ассоциации такими параметрами являются, в частности, тип и режим работы протокола защиты (AH или ESP), методы шифрования, секретные ключи, значение текущего номера пакета в ассоциации и другая информация. Наборы текущих параметров, определяющих все активные ассоциации, хранятся на обоих оконечных узлах защищенного канала в виде баз данных безопасных ассоциаций (SAD). Каждый узел IPSec поддерживает две базы SAD — одну для исходящих ассоциаций, другую для входящих.

Другой тип базы данных — база данных политики безопасности (SPD) — определяет соответствие между IP-пакетами и установленными для них правилами обработки. Записи SPD состоят из полей двух типов — полей селектора пакета и полей политики защиты для пакета с данным значением селектора (рис. 24.38).

Селектор в SPD включает следующий набор признаков, на основании которых можно с большой степенью детализации выделить защищаемый поток:

- IP-адреса источника и приемника могут быть представлены как в виде отдельных адресов (индивидуальных, групповых или широковещательных), так и диапазонами адресов, заданными с помощью верхней и нижней границ либо с помощью маски;
- порты источника и приемника (то есть TCP- или UDP-порты);
- тип протокола транспортного уровня (TCP, UDP);
- имя пользователя в формате DNS или X.500;
- имя системы (хоста, шлюза безопасности и т. п.) в формате DNS или X.500.

Для каждого нового пакета, поступающего в защищенный канал, IPSec просматривает все записи в базе SPD и сравнивает значение селекторов этих записей с соответствующими полями IP-пакета. Если значение полей совпадает с каким-либо селектором, то над пакетом выполняются действия, определенные в поле политики безопасности данной записи. Политика предусматривает передачу пакета без изменения, отбрасывание или обработку средствами IPSec.

В последнем случае поле политики защиты должно содержать ссылку на запись в базе данных SAD, в которую помещен набор параметров безопасной ассоциации для данного пакета (на рис. 24.38 для исходящего пакета определена ассоциация SA3). На основании заданных параметров безопасной ассоциации к пакету применяется соответствующие протокол (на рисунке — ESP), функции шифрования и секретные ключи.

Если к исходящему пакету нужно применить некоторую политику защиты, но указатель записи SPD показывает, что в настоящее время нет активной безопасной ассоциации с требуемой политикой, то IPSec создает новую ассоциацию с помощью протокола IKE, помещая новые записи в базы данных SAD и SPD.

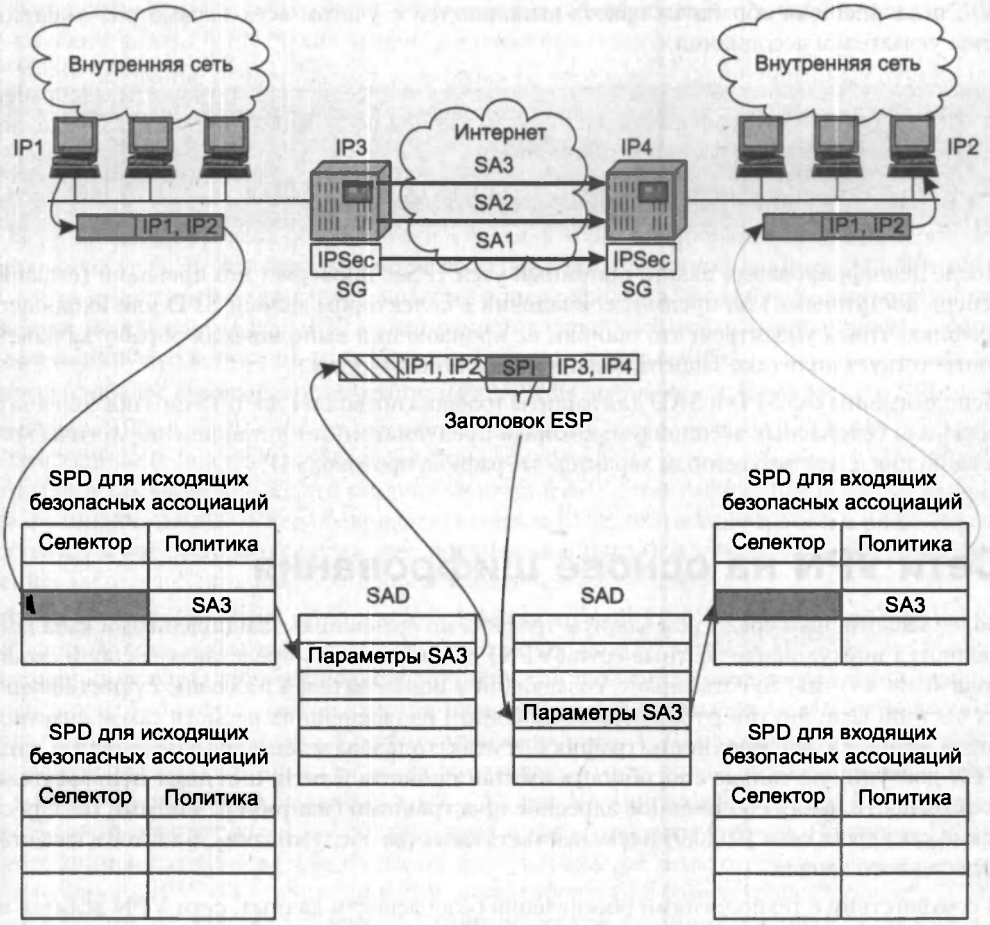


Рис. 24.38. Использование баз данных SPD и SAD

Базы данных политики безопасности создаются и администрируются либо пользователем (этот вариант больше подходит для хоста), либо системным администратором (вариант для шлюза), либо автоматически (приложением).

Ранее мы выяснили, что установление связи между исходящим IP-пакетом и заданной для него безопасной ассоциацией происходит путем селекции. Однако остается другой вопрос: как *принимающий* узел IPSec определяет способ обработки прибывшего пакета, ведь при шифровании многие ключевые параметры пакета, отраженные в селекторе, оказываются недоступными, а значит, невозможно определить соответствующую запись в базах данных SAD и SPD и, следовательно, тип процедуры, которую надо применить к поступившему пакету? Именно для решения этой проблемы в заголовках AH и ESP предусмотрено поле SPI. В это поле помещается указатель на ту строку базы данных SAD, в которой записаны параметры соответствующей безопасной ассоциации. Поле SPI заполняется протоколом AH или ESP во время обработки пакета в отправной точке защищенного канала. Когда пакет приходит в конечный узел защищенного канала, из его внешнего заголовка ESP или AH (на рисунке — из заголовка ESP) извлекается значение

SPI, и дальнейшая обработка пакета выполняется с учетом всех параметров заданной этим указателем ассоциации.

Таким образом, для распознавания пакетов, относящихся к разным безопасным ассоциациям, используются:

- на узле-отправителе — селектор;
- на узле-получателе — индекс параметров безопасности (SPI).

После дешифрирования пакета приемный узел IPSec проверяет его признаки (ставшие теперь доступными) на предмет совпадения с селектором записи SPD для входящего трафика, чтобы убедиться, что ошибки не произошло и выполняемая обработка пакета соответствует политике защиты, заданной администратором.

Использование баз SPD и SAD для защиты трафика позволяет достаточно гибко сочетать механизм безопасных ассоциаций, который предусматривает установление логического соединения, с дейтаграммным характером трафика протокола IP.

Сети VPN на основе шифрования

Более масштабным средством защиты трафика по сравнению с защищенными каналами являются виртуальные частные сети (VPN). Подобная сеть представляет собой своего рода «сеть в сети», то есть сервис, создающий у пользователей иллюзию существования их частной сети внутри публичной сети. Одним из важнейших свойств такой «частной сети» является защищенность трафика от атак пользователей публичной сети. Сетям VPN доступна не только способность имитации частной сети; они дают пользователю возможность иметь собственное адресное пространство (например, частные IP-адреса, такие как адреса сети 10.0.0.0) и обеспечивать качество обслуживания, близкое к качеству выделенного канала.

В соответствии с технологиями обеспечения безопасности данных, сети VPN делятся на два класса:

- *сети VPN на основе разграничения трафика* подробно обсуждались в разделе «Виртуальные частные сети» главы 19;
- *сети VPN на основе шифрования* работают на основе рассмотренной нами в предыдущем разделе техники защищенных каналов.

Виртуальная частная сеть на основе шифрования может быть определена как совокупность защищенных каналов, созданных предприятием в открытой публичной сети для объединения своих филиалов.

То есть в VPN техника защищенных каналов применяется уже в других масштабах, связывая не двух пользователей, а произвольное количество клиентских сетей.

Технологии VPN на основе шифрования включают шифрование, аутентификацию и туннелирование.

- Шифрование гарантирует конфиденциальность корпоративных данных при передаче через открытую сеть.

- Аутентификация отвечает за то, чтобы взаимодействующие системы (пользователи) на обоих концах VPN были уверены в идентичности друг друга.
- Туннелирование предоставляет возможность передавать зашифрованные пакеты по открытой публичной сети.

Для повышения уровня защищенности виртуальных частных сетей технологии VPN на основе шифрования можно применять *совместно* с технологиями VPN на основе разграничения трафика. Технологии VPN на основе разделения трафика иногда критикуют за недостаточный уровень безопасности, считая, что без шифрования трафика персонал поставщика услуг может получить несанкционированный доступ к данным. Действительно, такая вероятность существует, поэтому клиент услуг VPN на основе разграничения трафика, например MPLS VPN, может самостоятельно повысить защищенность своего трафика, прибегнув, скажем, к шифрованию передаваемых данных.

Сейчас наиболее широко используются сети VPN на основе протоколов IPSec и SSL.

Стандарты IPSec обеспечивают высокую степень гибкости, позволяя выбрать нужный режим защиты (с шифрованием или только с обеспечением аутентичности и целостности данных), а также использовать различные алгоритмы аутентификации и шифрования. Режим инкапсуляции IPSec позволяет изолировать адресные пространства получателя (клиента) и поставщика услуг за счет применения двух IP-адресов — внешнего и внутреннего.

Сети VPN на основе IPsec, как правило, строятся по типу CPVPN, то есть как виртуальные частные сети, в которых клиент самостоятельно создает туннели IPSec через IP-сеть поставщика услуг. Причем от последнего требуется только предоставление стандартного сервиса по объединению сетей, а значит, предприятию доступны как услуги сети поставщика, так и услуги Интернета. Конфигурирование сетей VPN на основе IPSec довольно трудоемко, поскольку туннели IPSec двухточечные, то есть при полносвязной топологии их количество пропорционально $N \times (N - 1)$, где N — число соединений. Необходимо учесть еще и непростую задачу поддержания инфраструктуры ключей. Протокол IPSec может применяться также для создания виртуальных частных сетей, поддерживаемых провайдером (PPVPN) — туннели в них также строятся на базе устройств клиента (CE-based), но эти устройства удаленно конфигурируются и администрируются поставщиком услуг.

Пропускная способность каналов и другие параметры QoS этой технологией не поддерживаются, но если оператор предоставляет определенные параметры QoS (например, за счет дифференцированного обслуживания), их можно использовать при создании туннеля IPSec.

В самое последнее время выросла популярность VPN на основе протокола SSL. Напомним, что этот протокол работает на уровне представления, непосредственно под уровнем приложений, так что приложения должны явным способом его вызывать, чтобы создать защищенный канал для своего трафика. Наиболее популярным приложением, использующим защищенные каналы SSL, является веб-браузер. В этом случае защищенные каналы SSL задействует протокол HTTP, и в этом режиме работы его часто называют протоколом HTTPS. Пользователи Интернета хорошо знают этот режим, так как браузер прибегает к нему во всех случаях, когда необходимо обеспечить конфиденциальность передаваемой информации: при покупках в интернет-магазинах, при интернет-банкинге и т. п.

Служба VPN на основе SSL функционирует на основе веб-портала, развернутого в локальной сети организации. Пользователи такой защищенной службы VPN получают удаленный

доступ к ресурсам этой локальной сети, обращаясь к веб-порталу посредством обычного браузера через порт 443 (TCP-порт протокола HTTPS). Отсутствие специального клиентского программного обеспечения, требующего настройки, является значительным преимуществом VPN на основе SSL.

Выводы

Информационная система находится в состоянии защищенности, если обеспечены ее конфиденциальность, доступность и целостность.

Информационная безопасность обеспечивается техническими средствами — системами шифрования, аутентификации, авторизации, аудита, антивирусной защиты, межсетевыми экранами и др., а также юридическими и морально-этическими нормами, просветительной работой и административными мерами.

Существует два класса алгоритмов шифрования — симметричные (например, DES) и асимметричные (например, AFS). Дайджест — это результат односторонней функции шифрования. Знание дайджеста не позволяет и даже не предполагает восстановления исходных данных. Дайджест используется для контроля целостности и аутентичности документа (цифровая подпись).

Аутентификация пользователя — это процедура доказательства пользователем того, что он есть тот, за кого себя выдает. Процедуры аутентификации могут основываться на знании разделяемого секрета (многоразовые и одноразовые пароли), владения неким уникальным предметом (физическим ключом, документом, сертификатом), на биохарактеристиках (рисунок радужной оболочки глаза).

Авторизация — это процедура контроля доступа легальных пользователей к ресурсам системы и предоставление каждому из них именно тех прав, которые определены ему администратором.

Антивирусная защита служит для профилактики и диагностики вирусного заражения, а также для восстановления работоспособности пораженных вирусами информационных систем. В ней используются методы, основанные на анализе содержимого файлов (сканирование сигнатур) и поведения программ (протоколирование и предупреждение подозрительных действий).

Сетевой экран осуществляет информационную защиту одной части компьютерной сети от другой путем анализа проходящего между ними трафика. Сетевые экраны делятся на экраны с фильтрацией пакетов на основе IP-адресов, сетевые экраны сеансового уровня, способные фильтровать пакеты с учетом контекста, и наиболее интеллектуальные сетевые экраны прикладного уровня.

Прокси-сервер — это особый тип приложения, которое выполняет функции посредника между клиентскими и серверными частями распределенных сетевых приложений, причем предполагается, что клиенты принадлежат внутренней (защищаемой) сети, а серверы — внешней (потенциально опасной) сети.

Технология защищенного канала обеспечивает защиту трафика между двумя точками в открытой транспортной сети, например в Интернете. Защищенный канал подразумевает выполнение трех основных функций:

- взаимная аутентификация абонентов при установлении соединения;
- шифрование передаваемых по каналу сообщений;
- подтверждение целостности поступающих по каналу сообщений, например, путем передачи одновременно с сообщением его дайджеста.

К числу наиболее популярных протоколов защищенного канала относятся IPsec и SSL. IPSec — это согласованный набор открытых стандартов, ядро которого составляют три протокола:

- AH гарантирует целостность и аутентичность данных;
- ESP, кроме того, обеспечивает конфиденциальность данных;
- IKE решает задачу автоматического распределения секретных ключей, необходимых для работы протоколов аутентификации.

Более масштабным средством защиты трафика по сравнению с защищенными каналами являются виртуальные частные сети (VPN). VPN на основе шифрования включают шифрование, которое гарантирует конфиденциальность корпоративных данных при передаче через открытую сеть, аутентификацию взаимодействующих систем на обоих концах VPN и туннелирование, позволяющее передавать зашифрованные пакеты по открытой публичной сети.

Вопросы и задания

1. В каких средствах обеспечения безопасности используется шифрование? Варианты ответов:
   а) аутентификация и авторизация;
   б) антивирусные системы;
   в) защищенный канал;
   г) сетевой экран прикладного уровня;
   д) фильтрующий маршрутизатор;
   е) цифровая подпись.
2. Какие из антивирусных методов способны обнаружить еще неизвестный вирус? Варианты ответов:
   а) сканирование сигнатур;
   б) метод контроля целостности;
   в) отслеживание поведения команд;
   г) эмуляция тестируемых программ.
3. К числу базовых функций сетевого экрана относятся:
   а) аудит;
   б) шифрование трафика;
   в) фильтрация трафика;
   г) антивирусная защита;
   д) функция прокси-сервера;
   е) авторизация;
   ж) повышение пропускной способности канала.
4. Существует ли угроза похищения пароля при использовании аппаратного ключа?
5. Справедливо ли утверждение «Поскольку открытый ключ не является секретным, то его не нужно защищать»?
6. Что содержится в электронном сертификате? Варианты ответов:
   а) секретный ключ владельца данного сертификата;
   б) данные о владельце сертификата;
   в) информация о сертифицирующем центре, выпустившем данный сертификат;
   г) зашифрованные открытым ключом сертифицирующего центра данные, содержащиеся в сертификате.

7. Правила доступа узлов сети периметра к ресурсам внутренней сети часто бывают более строгими, чем правила, регламентирующие доступ к этим ресурсам внешних пользователей. Как вы думаете, почему?
8. Какие из следующих утверждений верны:
   а) любое приложение после соответствующего конфигурирования имеет возможность работать через прокси-сервер;
   б) для работы через прокси-сервер приложение, изначально не рассчитанное на работу через проки-сервер, требует изменения исходного кода;
   в) каждое приложение, построенное в архитектуре клиент-сервер, непременно должно работать через прокси-сервер.
9. Почему в семействе протоколов IPSec функции обеспечения целостности и аутентичности данных дублируются в двух протоколах — AH и ESP?
10. Отметьте в таблице все возможные комбинации режимов работы протокола IPsec.

	Хост-хост	**Шлюз-шлюз**	**Хост-шлюз**
Транспортный режим			
Туннельный режим			

Ответы на вопросы

Глава 1

1. От вычислительной техники компьютерными сетями были унаследованы интеллектуальные возможности конечных узлов — компьютеров, а от телекоммуникационных сетей — методы передачи информации на большие расстояния.
2. Вычислительные ресурсы многотерминальных систем централизованы, а в компьютерной сети они распределены.
3. Значимые практические результаты по объединению компьютеров с помощью глобальных связей впервые были получены в конце 60-х годов.
4. Сеть ARPANET, созданная в конце 60-х, стала прародительницей Интернета.
5. Вариант а).
7. Технология Ethernet была стандартизована в 1980 году.
8. Компьютерные и телекоммуникационные сети сближаются в отношении типов услуг и используемых технологий.

Глава 2

2. Варианты б) и в).
4. Варианты б), ж) и з).
5. К сетевым службам относятся служба WWW, электронная почта, файловая служба, IP-телефония, справочная служба, DNS, DHCP, система управления сетью. Последние четыре ориентированы на администратора сети. Файловая служба, справочная служба, DNS, DHCP часто входят в состав сетевой ОС.
6. Варианты а), в) и г).

8. Рисунок 2.9, *а*, слева направо: ячеистая топология/звезда/дерево, полносвязная топология/кольцо. Рисунок 2.9, *б*, верхний ряд слева направо: полносвязная топология, ячеистая топология, ячеистая топология/звезда/дерево. Рисунок 2.9, *б*, нижний ряд слева направо: ячеистая топология/кольцо, ячеистая топология, ячеистая топология.
9. Вариант в).

Глава 3

1. Вариант г).
2. Варианты а) и г).
3. Дискретизация по времени соответствует частоте квантования амплитуды звуковых колебаний 1/25мкс, или 40 000 Гц. Для кодирования 1024 градаций звука требуется 10 двоичных разрядов. Отсюда необходимая пропускная способность для передачи оцифрованного таким образом голоса равна 40 000 × 10 = 400 Кбит/с.
4. Вариант б).
5. Время передачи данных увеличится примерно на 240 мс (детали см. на сайте www.olifer.co.uk).

Глава 4

1. Модель OSI стандартизует, во-первых, семиуровневое представление средств взаимодействия систем в сетях с коммутацией пакетов, во-вторых, перечень функций каждого уровня, в-третьих, названия всех уровней.
2. Да, модель взаимодействия открытых систем можно представить с другим количеством уровней, например, в модели TCP/IP определено только 4 уровня.
3. Все утверждения верны.
4. Модель OSI не рассматривает средства взаимодействия приложений конечных пользователей. Поэтому работа приложений *не может быть отнесена ни к одному из уровней модели OSI*. Однако некоторые приложения, вместо того чтобы обращаться к системным средствам организации сетевого взаимодействия, реализуют их «собственными силами». В таких случаях можно говорить о том, что приложение работает на соответствующем уровне (уровнях) модели OSI.
5. Как правило, протоколы транспортного уровня устанавливаются на конечных узлах. На промежуточных узлах сети, в частности на маршрутизаторах, транспортный протокол может быть установлен для поддержки дополнительных функций, например для удаленного управления промежуточным узлом, так как при этом промежуточный узел по отношению к управляющему узлу является конечным узлом.
6. Сетевые службы работают на прикладном уровне.
8. Никакие из перечисленных терминов синонимами не являются. К примеру, спецификация может быть как стандартизованной, так и не стандартизованной, а документ RFC может как являться, так и не являться стандартом.
9. Вариант г).

10. Да, компьютеры будут функционировать нормально. Отличие межуровневых интерфейсов в стеке протоколов двух компьютеров не помешает их сетевому взаимодействию.

11 и 12. Соответствующую информацию можно найти на сайтах www.ietf.org и www.rfc-editor.org.

Глава 5

5. С одной стороны, сеть оператора связи назвать корпоративной сетью нельзя, поскольку существует традиционное деление сетей на эти два типа. С другой стороны, можно, так как эта сеть может выполнять внутрикорпоративные функции, если принадлежит корпорации, которая занимается предоставлением телекоммуникационных услуг.
9. См. заполненную таблицу на сайте www.olifer.co.uk.
12. Поставщик услуг Интернета (ISP), поставщик интернет-контента (ICP), поставщик услуг хостинга (HSP), поставщик услуг по доставке контента (CDP), поставщик услуг по поддержке приложений (ASP), поставщик биллинговых услуг (BSP).

Глава 6

1. Да, краткосрочные и долгосрочные значения одной и той же характеристики могут различаться.
3. Для пакета фиксированной длины фиксированными являются время сериализации и время задержки пакета.
4. От длины пакета зависит время его сериализации.
5. Варианты а), б) и в).
6. Медиана равна 17 мс, среднее значение — 1441,7.
7. Задержки в сети лучше характеризует медиана, так как ее значение ближе к значениям большинства задержек выборки.
8. 85-процентный квантиль равен 20 мс, так как задержки шести пакетов (85 %) меньше или равны 20 мс.
10. Единичное значение односторонней задержки пакета зависит от размера пакета, так как задержка измеряется между моментом помещения в исходящую линию связи *первого* бита пакета узлом-отправителем и моментом приема *последнего* бита пакета с входящей линии связи узла-получателя.
12. Вариацию задержки можно компенсировать применением буфера достаточного размера.
13. Избирательная функция формирует пары пакетов, для которых вычисляется разность односторонних задержек.
14. Варианты а) и б).
16. Трафик может передаваться с большими задержками, но без джиттера, например, при передаче по спутниковому каналу задержки для всех пакетов велики, но одинаковы.

Глава 7

1. В сетях с коммутацией каналов очереди не возникают.
2. На размер очереди в наибольшей степени влияет коэффициент загрузки.
3. Приоритетное обслуживание не дает никаких гарантий в отношении средней пропускной способности для трафика очередей более низких приоритетов.
4. В отношении предсказуемости скорости передачи данных приложения можно разделить на приложения с потоковым и пульсирующим трафиком.
5. При увеличении пульсации потока задержки, связанные с пребыванием пакетов этого потока в очереди, увеличатся.
6. Обслуживающий прибор модели M/M/1 обычно соответствует выходному интерфейсу маршрутизатора, при этом производительность обслуживающего прибора равна пропускной способности интерфейса.
7. Причиной возможного возникновения очередей в сети с коммутацией пакетов даже при невысокой средней загрузке коммутаторов и маршрутизаторов являются значительные кратковременные перегрузки.
8. Вариант б), так как трафик загрузки больших файлов данных требует некоторой гарантированной пропускной способности, обеспечиваемой при взвешенном обслуживании, и не чувствителен к задержкам, которые могут возникать при таком обслуживании.
9. Комбинировать приоритетное и взвешенное обслуживание можно. В наиболее популярном алгоритме подобного рода поддерживается одна приоритетная очередь и несколько очередей, обслуживаемых в соответствии с взвешенным алгоритмом.
10. Второй поток будет испытывать в очереди наименьшие задержки, так как он должен обслуживаться при относительном коэффициенте использования 0,5 — это минимальный коэффициент для всех потоков.
11. Вариант б).
12. Да, в те периоды, когда скорость потока *A* оказывается меньше зарезервированной для этого потока пропускной способности, эта пропускная способность может использоваться потоком *B*.
13. При инжиниринге трафика меняется маршрут.
14. Варианты б) и в).
16. При работе сети в недогруженном режиме операторы обычно выполняют мониторинг коэффициента использования пропускной способности линий связи сети.

Глава 8

1. Термин «линия связи» является синонимом всех трех представленных терминов.
3. Цифровой канал может передавать аналоговые данные, если они оцифрованы.
4. Усилители только увеличивают мощность сигнала, в то время как регенераторы помимо увеличения мощности восстанавливают исходную форму сигнала.
5. Теоретически спектр сигнала некоторой определенной формы можно найти с помощью формул Фурье, а сделать это практически можно с помощью спектрального анализатора.

6. Вариант в).
7. Варианты а), в) и г).
8. Варианты а) и б).
10. Для устойчивой передачи данных мощность передатчика в 40 дБм достаточна, так как кабель вносит затухание $-0{,}2 \times 60 = -12$ дБ, а это снижает мощность сигнала на входе до 28 дБм, что выше порога приемника в 20 дБм.
11. Причиной перекрестных наводок на ближнем конце кабеля является влияние электромагнитного поля, создаваемого передатчиками, на соседние провода кабеля, к которым подключены входы приемников.
12. Повысить пропускную способность канала за счет увеличения числа состояний информационного сигнала удается не всегда, поскольку это может привести к выходу спектра за пределы полосы пропускания линии.
13. Помехи в кабелях UTP подавляются за счет скручивания проводов.
14. Более качественно передает сигналы кабель с большим по абсолютной величине значением NEXT.
15. Для передачи данных на большие расстояния предназначен одномодовый кабель.
16. Вариант в), так как значения импеданса передатчика и кабеля будут не совпадать.
17. Теоретический предел скорости передачи данных рассчитывается следующим образом:

$$C = F\log_2(1 + P_с/P_ш) = 1\,000\,000 \times \log_2(1 + 62/2) = 1\,000\,000 \times \log_2(32) = 5 \text{ Мбит/с}.$$

Глава 9

1. В методе BFSK используется две частоты.
2. Вариант а).
3. Пятый бит добавляется для устойчивого распознавания 4-х информационных битов при искажении сигналов.
4. Количество битов, которое передает один символ кода, имеющий 10 состояний, рассчитывается по следующей формуле:

$$\log_2 10 = 3{,}32.$$

6. Варианты б) и в).
7. Для улучшения самосинхронизации кода B8ZS применяется искусственное искажение последовательности нулей запрещенными символами.
8. Варианты а) и б).
11. Варианты а) и в).
12. Вариант а).
13. Вариант б).
14. В схемах контроля по паритету расстояние Хемминга равно 2.
16. В сетях с коммутацией пакетов используется асинхронный режим.
17. Первыми двумя гармониками являются 25 МГц и 75 МГц.

19. Нет, данные по каналу надежно передаваться не могут. Полоса пропускания равна 1 МГц, спектр с учетом первых двух гармоник — 10 МГц, что значительно шире имеющейся полосы пропускания.
20. Учитывая частоту появления символов, можно выбрать следующие коды: O — 1, A — 01, D — 001, B — 001, C — 0001, F — 00000. В этой кодировке для передачи указанного сообщения потребуется 35 бит. Таким образом, достигается компрессия по сравнению с обоими случаями. Кодировка ASCII дает 128 бит. В случае использования кодов равной длины при наличии только данных шести символов для кодирования одного символа достаточно 3 бит, а для всего сообщения — 48 бит.
21. Ширина спектра увеличится в два раза.

Глава 10

3. Вариант а).
4. Радиоволны с частотами от 2 до 30 МГц могут распространяться на сотни километров за счет отражения ионосферой Земли.
5. Для спутниковой связи используется спектр 1,5–30,5 ГГц.
6. Распространению микроволн мешают туман, роса, дождь.
7. Вариант б).
8. Вариант б).
9. Эллиптические орбиты позволяют обеспечить связью районы, близкие к Северному и Южному полюсам.
10. Варианты а), б) и г).
11. Технология FHSS является высокоскоростной при условии, что применяется высокоскоростной метод кодирования для каждой из частот.
12. Последовательность Баркера используется в технологии DSSS благодаря ее свойству быстрой синхронизации приемника с передатчиком.
13. Основным свойством расширяющих последовательностей, используемых в технологии CDMA, является взаимная ортогональность кодов.
14. Нет, указанные последовательности использовать нельзя, так как сами последовательности и их инверсии не являются ортогональными относительно операций, определенных для сигналов DSSS.
15. 01001000111.

Глава 11

2. Варианты а), б) и г).
4. Нет, в сети PDH нельзя выделить канал DS-0 непосредственно из канала DS-3.
5. Вместо «кражи бита», применяемой в канале T-1, в канале E-1 выделяют для служебных целей два байта, нулевой и шестнадцатый.
8. Отсутствие синхронности трибутарных потоков компенсируется за счет «плавающих» виртуальных контейнеров внутри кадра SDH.

9. Кадр STM-1 может мультиплексировать 63 канала.
10. Кадр STM-1, если в нем уже мультиплексировано 15 каналов E-1, может мультиплексировать 64 канала.
11. В кадре STM-1 используется три указателя, так как он может содержать три различных виртуальных контейнера уровня VC-3.
13. Вариант б).
14. Защита MS-SPRing более эффективна, чем SNC-P, если трафик распределяется между мультиплексорами сети равномерно.
15. Вариант б).
16. Нет, объединять контейнеры VC-3 за счет смежной конкатенации нельзя.
17. Да, составляющие контейнеры при виртуальной конкатенации можно передавать по разным маршрутам.
18. Да, пропускную способность соединения SDH можно изменить динамически, если в сети работает механизм LCAS.
19. Протокол GFP в режиме GFP-F не использует для выравнивания скоростей пустые кадры, потому что в этом режиме кадры полностью буферизуются.
20. И в сетях FDM, и в сетях DWDM используется частотное мультиплексирование.
22. В сетях DWDM регенераторы служат для устранения нелинейных искажений оптического сигнала.
23. Причиной ухудшения качества оптического сигнала является его хроматическая дисперсия.
24. Операция выравнивания выполняется, когда разница в принятых и переданных данных составляет 3 байта. Каждую секунду разница составляет $10{-}5 \times 155 \times 10{+}6 = 1550$ бит, поэтому частота отрицательного выравнивания равна $1550/24 = 64{,}58$ Гц.
25. Варианты б) и в).

Глава 12

1. Варианты а) и г).
3. Варианты б) и в).
4. Вариант б).
5. Преамбула и начальный ограничитель кадра в стандарте Ethernet служат для входа приемника в побайтный и побитный синхронизм с передатчиком.
6. Вариант б).
8. Скорость передачи пользовательских данных равна 9,597 Мбит/с.
9. Варианты а) и г).
10. Вариант а).
11. Время равно 368 мс (детали см. на сайте www.olifer.co.uk).
13. Вариант б).
14. Вариант в).

15. Вариант б).
16. Вариант в).
18. Да, станция может передать кадр через точку доступа.
19. Вариант в).
20. Режим PCF всегда имеет приоритет перед режимом DCF, поскольку межкадровый интервал в режиме PCF меньше, чем в DCF.
21. Вариант в).

Глава 13

1. Варианты а) и в).
2. Вариант б).
3. Вариант б).
4. Правильны все варианты.
5. Записи таблицы продвижения имеют ограниченный срок жизни с целью динамического и автоматического отражения изменений топологии сети.
6. Нет, скорость продвижения не может превосходить скорость фильтрации.
7. Варианты а), в) и г).
8. Варианты б) и в).
9. Вариант б).
10. Варианты а) и б).
11. Да, форматы кадров 10 Мбит/с Ethernet и Fast Ethernet совпадают.
12. Вариант в).
13. Нет, в технологии 10G Ethernet разделяемая среда не используется.
14. Нет, для волоконно-оптических портов режим автопереговоров не поддерживается.
15. Цифра 4 говорит о том, что информация в каждом направлении передается с помощью четырех волн.
16. Нет, если только мультиплексор не имеет специальный порт 10GBase-WL.

Глава 14

1. Варианты а) и в).
2. Нет, корневой мост не имеет корневых портов.
3. Вариант а).
4. Да, администратор может влиять на выбор корневого коммутатора, задавая значения старших двух байтов идентификатора коммутаторов.
5. Выбор активной топологии завершается через определенное время.
6. Варианты б), в) и г).
8. Вариант б).

9. Варианты а), б) и в).
10. Варианты б) и в).
11. Варианты б), в) и г).
12. Группирование портов плохо работает в сети, построенной на нескольких коммутаторах, из-за слишком больших накладных расходов: для соединения коммутаторов нужно использовать столько портов, сколько сетей VLAN существует в сети.
13. Да, можно одновременно использовать группирование портов и стандарт IEEE 802.1Q.
14. Да, алгоритм покрывающего дерева должен учитывать наличие в сети VLAN.

Глава 15

1. Варианты а) и в). Идентификатор виртуального канала и MAC-адрес могут являться локальными (аппаратными) адресами интерфейсов, если соответствующие сети включены в составную IP-сеть в качестве подсетей.
2. Варианты а) и г). Детали см. на сайте www.olifer.co.uk.
4. Номер подсети 108.5.16.0. Для нумерации интерфейсов в данной сети может быть использовано 12 бит, то есть 4096 значений. Но так как двоичные значения, состоящие из одних нулей и одних единиц, зарезервированы, то в сети не может быть более 4094 узлов.
5. Об IP-адресах узлов ничего определенного сказать нельзя (детали см. на сайте www.olifer.co.uk).
6. Вариант в).
7. Количество ARP-таблиц соответствует числу сетевых интерфейсов с назначенными IP-адресами.
9. При наличии DHCP-агентов достаточно одного DHCP-сервера.
10. Максимум можно организовать 16 385 подсетей. При этом маска должна иметь значение 255.255.255.252 (детали см. на сайте www.olifer.co.uk).
11. Администратор должен иметь 25 адресов при условии, что в сети установлен DHCP-сервер.

Глава 16

4. Вариант в).
5. Записей о маршрутах по умолчанию в таблице маршрутизации может быть несколько.
7. Нет, в IP-пакете маска не передается.
9. Вариант б).
10. Такое сочетание адреса сети и маски дает совпадение с любым IP-адресом.
11. Вариант г).
12. Вариант в).

Глава 17

1. Объем полученных данных составляет 165 005 байт.
2. Варианты а) и г).
3. Да, в сети можно обойтись без протоколов маршрутизации, если создавать таблицы маршрутизации вручную.
5. Вариант в).
6. Варианты а), б) и в).
7. Варианты а), б) и г).
8. Вариант в). ICMP-сообщение всегда направляется узлу-отправителю пакета, вызвавшего ошибку. Оно обрабатывается либо ядром операционной системы, либо протоколами транспортного и прикладного уровней, либо приложениями, либо просто игнорируется. Обработка ICMP-сообщений в функции протоколов IP и ICMP не входит.

Глава 18

3. Вариант г).
5. Варианты б) и в).
6. В качестве номеров назначенных портов могут выступать произвольные числа, уникальные для данного глобального IP-адреса, например 4100, 4102, 4103.
7. Вариант а).
9. Вариант в).

Глава 19

1. Вариант б).
2. Варианты а) и б).
3. Варианты б) и в).
4. Варианты б) и в).
5. При туннелировании роль несущего протокола чаще всего исполняет протокол IP.
6. Вариант б).
7. Да, были помечены кадры 6 и 7, так как согласованная величина пульсации равна: CIR × T = 51 200 бит = 6400 байт, и это значение превышается 6-м кадром.
8. Вариант а).
9. Вариант б).
10. Вариант в).
11. Варианты а) и в).
12. Варианты а), б) и в).
13. Вариант б).

Глава 20

1. Вариант в).
2. Варианты б) и в).
3. Максимальное число уровней иерархии путей LSP стандартами MPLS не ограничивается.
4. Да, в сети, поддерживающей MPLS, часть трафика можно передавать посредством обычного IP-продвижения.
5. Вариант в).
6. Вариант в).
7. Вариант а).
8. Вариант а).
9. Вариант б).
10. Вариант в).
11. Варианты б) и в).
12. Вариант в).
13. Варианты б) и в).

Глава 21

1. Правильны все варианты ответов.
2. Варианты б), в) и г).
3. Варианты б) и в).
4. Варианты б) и в).
5. Правильны все варианты ответов.
6. Варианты а) и б).
7. Максимальное количество псевдоканалов равно 1 048 576. Эта величина определяется разрядностью метки MPLS.
8. Нет, устройство PE не должно изучать MAC-адреса клиентов.
9. Вариант б).
10. Вариант б).
11. Да, стандарт Y.1731 дополняет функции стандарта CFM набором функций мониторинга производительности сети.
12. Вариант б).
13. Вариант в).

Глава 22

1. Варианты а), б) и в).
2. Принимать во внимание нужно набор дополнительных услуг, которыми клиенты хотели бы воспользоваться.
3. Универсальным можно назвать абонентское окончание, которое обеспечивает передачу всех видов трафика: компьютерного, телефонного и телевизионного.

4. Варианты а) и б).
5. При конфигурировании маршрутизаторов имеет место удаленное управление с помощью протокола telnet.
6. Варианты б) и в).
7. Это зависит от функциональности модема. Если модем не кадрирует информацию и оперирует только с потоком битов, то он является устройством физического уровня. Если же он кадрирует информацию, то это устройство канального уровня.
8. Вариант а).
9. Вариант в).
10. Соединение будет работать на скорости 33,6 Кбит/с.
11. Вариант а).
12. Вариант а).

Глава 23

2.

Используется почтовым клиентом для передачи письма на сервер	SMTP
Используется почтовым клиентом для получения письма с сервера	POP3, IMAP
При получении почты письмо перемещается с сервера на клиент	POP3
При получении почты письмо копируется с сервера на клиент	IMAP

3.

Путь к объекту	/mobile/web/versions.shtml
DNS-имя сервера	www.bbc.co.uk
URL-имя	http://www.bbc.co.uk/mobile/web/versions.shtml
Тип протокола доступа	http://

4. Варианты б), в) и е).
5. Варианты б), г) и д).
6. Варианты а), в) и г).
7. Варианты а), б) и г).

Глава 24

1. Варианты а), г), д) и е).
2. Варианты б), в) и г).
3. Варианты а), в) и д).
4. Да, при использовании аппаратного ключа существует угроза похищения пароля в течение интервала существования «разового» значения ключа.
5. Нет, это утверждение несправедливо, открытый ключ необходимо защищать от подмены.
6. Варианты б) и в).
7. Вариант б).
8. Все комбинации возможны, кроме работы в транспортном режиме защищенного канала, построенного по схеме шлюз-шлюз.

Рекомендуемая и использованная литература

1. *Фред Халсалл.* Передача данных, сети компьютеров и взаимосвязь открытых систем, М.: Радио и связь, 1995.
2. *Столингс В.* Передача данных, 4-е изд. СПб.: Питер, 2004.
3. *Столингс В.* Современные компьютерные сети, 2-е изд. СПб.: Питер, 2003.
4. *Куроуз Дж., Росс К.* Компьютерные сети, 4-е изд. СПб.: Питер, 2004.
5. *Таненбаум Э.* Компьютерные сети, 4-е изд. СПб.: Питер, 2002.
6. *Фейт Сидни.* TCP/IP. Архитектура, протоколы, реализация. М.: Лори, 2000.
7. *Стивен Браун.* Виртуальные частные сети. М: Лори, 2001.
8. *Шринивас Вегешна.* Качество обслуживания в сетях IP. М.: Вильямс, 2003.
9. *Аннабел З. Додд.* Мир телекоммуникаций. Обзор технологий и отрасли. М.: ЗАО «Олимп-Бизнес», 2002.
10. *Кеннеди Кларк, Кевин Гамильтон.* Принципы коммутации в локальных сетях Cisco. М.: Вильямс, 2003.
11. *Дуглас Э. Камер.* Сети TCP/IP. Том 1. Принципы, протоколы и структура. М.: Вильямс, 2003.
12. *Блэк Ю.* Сети ЭВМ: протоколы стандарты, интерфейсы. Перев. с англ. М.: Мир, 1990.
13. *Ричард Стивенс.* Протоколы TCP/IP. Практическое руководство. СПб.: БХВ-Санкт-Петербург, 2003.
14. *Слепов Н. Н.* Синхронные цифровые сети SDH. М.: Эко-Трендз, 1998.
15. *Денисьев и Мирошников.* Средства связи для «последней мили». М.: Эко-Трендз, 1998.
16. *Дилип Найк.* Стандарты и протоколы Интернета. М.: Channel Trading Ltd., 1999.
17. *Уолрэнд Дж.* Телекоммуникационные и компьютерные сети. Вводный курс, М.: Постмаркет, 2001.
18. *Гольдштейн Б. С., Пинчук А. В., Суховицкий А. Л.* IP-телефония. М.: Радио и связь, 2001.
19. *Олифер В. Г., Олифер Н. А.* Новые технологии и оборудование IP-сетей. СПб.: БХВ-Санкт-Петербург, 2000.
20. *Олифер В. Г., Олифер Н. А.* Сетевые операционные системы, 2-е изд. СПб.: Питер, 2008.

Алфавитный указатель

3Com 427
6Bone 648
10Base-2 367
10Base-5 367
10Base-T 367
10G Ethernet 436
100Base-FX 428, 430
100Base-T4 428
100Base-TX 428, 430
100VG-AnyLAN 427
1000Base-LX 434
1000Base-SX 434
Эрикссон Ларс Магнус 285

A

ABR 681
AC 737
ACK 559
ACL 203, 392
ACR 241
ADM 320
AES 850
AF 613
AH 892
AM 231, 257
AMI 266
AP 294
API 115
APS 325
Arcnet 32
ARP 61, 497
ARPANET 29
ARP-запрос 497
ARP-кэш 500
ARP-ответ 497
AS 587
ASK 257
ASM 626
ASP 160
ATM 280, 678
AU 318
AUG 319
AWG 339

B

Basic NAT 617
Bc 676
Be 676
BEB 748, 750
BECN 674
BER 242
BFD 717
BFSK 258
BGP 588
BGPv4 588
Bluetooth 390
BPDU 451
BPSK 258
BRI 778
BS 293
BSP 160
BSS 378

C

CA 865
CBR 187, 681

CCM 743
CD 364
CDMA 276, 306
CDP 159
CE 685
CET 730, 741
CGI 808
CHAP 690, 861
CIDR 494, 544
CIR 675
Cisco 600
CLNP 127
CMIP 826
CMTS 788
CO 150
CONP 127
CoS 704
CP 326
CPE 777
CPVPN 683
CRC 275
CS 362
CSMA/CA 380
CSMA/CD 362
CTS 383
CW 381

D

DCC 322
DCE 232
DCF 380
DDoS 832
DE 674
DES 849
DHCP 508
DHCP-агент 510
DiffServ 603
DIFS 383
DIX 358
DLCI 94, 674
DM 630
DMZ 541
DNS 488, 505
DNS-имя 488
DoS 832
DS 379
DSL 777
DSLAM 786
DSn 311
DSP 281, 435
DSS 379
DSSS 305
DTE 232
DTMF 770
DVA 573
DVMRP 632
DWDM 278, 310, 333
DXC 320

E

E-1 311
E-2 311
E-3 311
eBGP 590
EDR 394
EF 613
EGP 588
EHF 287
ELF 287
ENUM 818
EoMPLS 730
EOT 731
EPL 733
EPP 414
EPS 326
ESP 892
ESS 379
Ethernet 32, 34, 103
  на основе
    Ethernet 730
    MPLS 730
    транспорта 731
  операторского класса 727
Ethernet DIX 358
Ethernet II 358
EtherSwitch 414
EtherType 471
ETSI 317
EVC 732
EVPL 733

F

Fast Ethernet 32
FCS 168, 274, 361
FDD 281
FDDI 32
FDM 233, 276
FEC 275, 703
FECN 674
FEXT 240
FHSS 302
FIFO 192
FIN 559
FIND 829
FLP 431
FM 231, 257
FP 642
FPGA 438
FQDN 504
Frame Relay 34, 672
FSK 257

FTN 704
FTP 483, 819

G

G.703 314
GBIC 443
GEO 298
GFP 332
Gigabit Ethernet 32, 432
GPS 300

H

HDLC 690
HDWDM 335
HTML 802
HTTP 483, 805

I

IAB 126
IAM 781
IANA 627
iBGP 590
IBM 32, 33
ICANN 494
ICMP 484, 591
ICV 897
IDS 888
IEEE 802.1Q 469
IEEE 802.3ae 436
IETF 126
IGMP 627
IGP 588
IKE 892
IMAP 800
IN 36
INAP 816
Internet 29
intranet 34
IntServ 603, 604
IP 484
IPDV 177
IPG 363
IPPM 174
IPSec 646
IPv4-отображенный IPv6-адрес 647
IPv4-совместимый IPv6-адрес 647
IPX 128
IP-коммутация 699
IP-телефония 35
IRTF 126
ISDN 35, 775
IS-IS 127
ISM 291
ISO 113
ISOC 126
ISP 157, 159
ITU-T 113, 317
IVR 813
IX 159

J

jam-последовательность 364
JC 346

K

Kalpana 413

L

L2CAP 393
L2VPN 686
L3VPN 688
LAN 31, 139
LAP-M 775
LCAP 464
LCAS 332
LCN 94
LCP 690
LDP 701, 709
LED 251
LEO 298
LER 702
LHC 51
LIT 370
LLC 358
LLC1 358
LLC2 358
LLC3 358
LMDS 790
LOS 289
LSA 573, 583
LSP 702
LSR 698

M

MA 362
MAC 116, 358
MAC-адрес 59, 360, 486
MAN 35, 139
MEF 731
MEMS 341
MEO 298
MEP 743
MFSK 258
MG 815
MGCP 816
MIB 823
MII 429
MIMO 388
MIP 743
MIT 768

MLPPP 690
MMDS 790
MMF 250
MNP 775
MOSPF 635
MPLS 698
MSOH 321
MSP 327
MS-SPRing 329
MSTP 474
MTU 547

N

NAP 159
NAPT 617
NAPTR 818
NAT 616
NAVSTAR 300
NCP 128, 690
NE 821
NetBEUI 129
NetBIOS 129
NEXT 240
NFC 398
NGN 35
NIC 41
NJO 345
NLA 643
NM 517
NMS 821
NNI 748
Novell NetWare 32
nPersonNamertVBR 681
NPN 35
NRZI 266
NSP 738
NT 777
NT1 783
NT2 783

O

OADM 338
OAM 742
Och 344
OC-N 317
ODU 344
OFDM 302
OLE_LINK1приложение
интерактивное 189
OPU 344
OSI 108, 113
OSPF 582
OTN 310, 342
OTU 344
OUI 360
OWD 174
OXC 338

P

PAN 351, 389
PAP 690
PB 746
PBB 748
PBX 144
PCF 380
PCM 262
PDH 310
PDU 116
PersonNamertVBR 681
PHB 612
PHP 704
PHY 429
PIFS 383
PIM 635
PIM-DM 635
PIM-SM 635
PIN 864
PIR 178
PJO 345
PKI 869
PNNI 681
POH 318
POP 150
POP3 800
POS 689
PPP 690
PPTP 772
PPVPN 683
PRC 314
PRI 778
PS 240
PS FEXT 240
PSH 559
PSK 257
PS NEXT 240
PVC 680

Q

QAM 259
QoS 36

R

RAC 765
RARP 501
RAS 765, 810
RED 607
RFC 126
RFC 1700 555
RFC 2131 508
RFC 2132 508

RFC 3232 555
RIP 575
RP 631
RPF 632
RPT 636
RSA 854
RSOH 321
RST 559
RSTP 449
RSVP 604, 609
RSVP TE 721
RTP 809
RTS 382
RTT 176

S

SA 893
SAD 899
SAP 128
SCO 392
SCP 143
SCS 252
SDC 775
SDH 310, 317
SDH NG 331
SFF 443
SFP 443
SG 894
SGCP 816
SIFS 383
SIP 811
SIR 178
SLA 165, 643
SLIP 689
SM 630
SMB 130
SMF 250
SMS 822
SMTP 483, 795
SN 896
SNC-P 328
SOCKS 886
SONET 316
SPD 899
SPI 896
SPT 637
SPX 128
SRC 315
SS7 771
SSL 122, 891
STA 412, 449
STDM 280
STM 280
STM-N 317
STP 231, 449
Stratum 1 314
Stratum 2 315
STS-N 317
SVC 680
SYN 559
SynOptics 427

T

T-1 311
T-2 311
T-3 311
TA 783
TCP 483, 554
TCP-порт 555
TCP-сокет 556
TDD 281
TDM 233, 276, 278
TE 216, 718, 777
TE1 781
TE2 783
telnet 483, 767
TE-туннель 718
  свободный 718
  строгий 718
TLA 642
TM 320
Token Bus 32
ToS 515
TS 346
TTL 516, 547
TU 318

U

UBR 681
UDP 483, 554
UDP-дейтаграмма 557
UDP-порт 555
UDP-сокет 556
UHF 231
UNI 732
URG 559
URL 803
UTP 231

V

V.42 774
VBR 188
VC 95, 318
VCI 94
VHF 231
VLAN 467
VPLS 733, 739
VPN 148, 662
VPWS 733, 737
VSAT 299

W

WAN 28
WAP 804
WDM 276, 334
WEP 384
WFQ 202
WLAN 375
WLL 288, 790
WPA 385
WRED 608
WWW 34, 801

X

X.25 29
XGMII 436
XID 775

А

абонент 78
абонентское окончание 144
  беспроводное 790
  проводное 790
Абрамсон Норман 354
абсолютный уровень мощности 237
автоматическое защитное переключение 325
автоматическое назначение
  динамических адресов 509
  статических адресов 509
автономная система 587
автопереговоры 430
авторизация 859
агент пользователя 803
агрегатный порт 319
агрегирование
  адресов 540, 644
  линий связи 459
  физических каналов 459
агрегированный поток 615
адаптер
  сетевой 41
  терминальный 783
адаптивная компрессия 272
адаптивная маршрутизация 573
административный блок 318
администратор 528
адрес
  IP-адрес 487
  MAC-адрес 59, 486
  агрегируемый 642
  аппаратный 60, 486
  виртуального интерфейса 519
  выходного интерфейса 520
  глобальный 120, 642
  групповой 59, 360, 490, 491, 528
  индивидуальный 360, 490
адрес (*продолжение*)
  локальный 486
  назначения
    пакета 520
    потока данных 63
  неопределенный 491
  обратной петли 491, 519
  ограниченный 491
  особого назначения 527
  порта 526
  произвольной рассылки 59
  разрешение 61
  сетевой 120, 487
  символьный 59
  следующего маршрутизатора 520, 526
  уникальный 59, 642
  частный 493, 616
  числовой 59
  широковещательный 59, 360, 491, 528
адресация 59
  иерархическая 60
  плоская 60
адресная таблица 407, 408
адресное пространство 59
активное измерение 171
активное сопротивление 239
активное управления очередями 206
алгоритм
  FIFO 197
  адаптивной маршрутизации 573
  ведра маркеров 605
  взвешенных очередей 200
  Дийкстры 583
  динамической маршрутизации 573
  дистанционно-векторный 574
  комбинированный 202
  покрывающего дерева 412, 449
  приоритетного обслуживания 197
  прозрачного моста 406, 449
  состояния связей 573, 574, 719
  Хафмана 273
  широковещания и усечения 634
  шифрования 854
альтернативный порт 457
амплитудная манипуляция 257
амплитудная модуляция 257, 258
анализ
  надежности 822
  производительности 822
аналоговая линия связи 233
аналоговый телефон 770
аналого-цифровой преобразователь 261
антенна
  изотропная 287
  направленная 287
  ненаправленная 287
  параболическая 287

антивирусная защита 872
аппаратный адрес 60, 486
аппаратный ключ 862
аппаратура
передачи данных 232
промежуточная 232
арбитр 71
аренда
IP-адресов 509
каналов 148
арендуемый канал 662
асинхронное отображение нагрузки 345
асинхронное приложение 189
асинхронный канал 392
асинхронный режим
временного мультиплексирования 278
передачи 280, 678
атака
отказа в обслуживании 832
понятие 831
распределенная 832
атакующий блок 839
аудиоуровень 394
аудит 860
аукцион 291
аутентикод 872
аутентификация
данных 859
пользователя 857
понятие 870
приложений 858
строгая 858
устройств 859
АЦП 261

Б

база данных
безопасных ассоциаций 899
политики безопасности 899
управляющей информации 823
базовая станция 293
базовая трансляция сетевых адресов 617
базовый набор услуг 378
байт дифференцированного обслуживания 515
баланс нагрузки 91
Баркера последовательность 305
бастион 882
безопасная ассоциация 893
безопасность
информационная 830, 846
транспортных услуг 164
бесклассовая междоменная маршрутизация 494, 544
беспроводная локальная сеть 375
беспроводная связь
мобильная 285
беспроводная связь (*продолжение*)
фиксированная 285
беспроводная сеть 140
беспроводная среда 230, 287
беспроводное абонентское окончание 790
биполярное кодирование с альтернативной инверсией 266
биполярный импульсный код 267
БИС 30
бит
кодовый 559
синхронизации 312
битовая скорость
передатчика 54
переменная 188
постоянная 187
битовый интервал 365, 429
бит-стаффинг 313
блок
административный 318
атакующий 839
данных оптического канала 344
коммутирующий 652
поиска целей 839
пользовательских данных 344
транспортный оптического канала 344
трибутарный 318
управления
жизненным циклом 843
удаленного 842
центральный 652
фиксации событий 843
бод 245
большая интегральная схема 30
большой адронный коллайдер 51
брандмауэр 876
браузер 803
буфер 88
буферная память 88
быстрое продвижение 613
быстрое расширение спектра 303
быстрый протокол покрывающего дерева 449

В

вариация задержки пакета 177
веб-браузер 46
веб-документ 802
веб-клиент 803
веб-сервер 804
веб-страница 802
ведро маркеров 605
вектор атаки 839
величина пульсации 178
дополнительная 676
согласованная 676
вероятность отказа 179
вертикальная подсистема 252

вертикальный контроль по паритету 274
взаимная идентификация 775
взаимодействие
межсетевое 118
открытых систем 108, 113
взвешенная очередь 200
взвешенное обслуживание 200, 202
взвешенный алгоритм RED 608
ВЗГ 315
видимый свет 288
виртуальная конкатенация 331
виртуальная локальная сеть 467
виртуальная частная линия Ethernet 733
виртуальная частная сеть 148, 662, 682, 901
на базе
инфраструктуры поставщика 683
оборудования потребителя 683
поддерживаемая
клиентом 683
поставщиком 683
виртуальное соединение 681, 732
виртуальный канал 94
двунаправленный 673
коммутируемый 680
однонаправленный 673
постоянный 673, 680
виртуальный контейнер 318
виртуальный путь 681
вирус 843
витая пара
категории 3 244
категории 5 236
неэкранированная 231, 247, 248
понятие 247
экранированная 231, 247, 249
внешний шлюз 587
внешний шлюзовой протокол 588
внешняя помеха 235
внешняя сеть 684
внешняя угроза 831
внутренний шлюзовой протокол 588
внутренняя помеха 235
внутренняя сеть 684
возможность
отрицательного выравнивания 345
положительного выравнивания 345
волновое мультиплексирование 276, 277, 334
высокоуплотненное 335
уплотненное 333
волновое сопротивление 239
волокно
выделенное 667
многомодовое 250
одномодовое 250
оптическое 667
темное 667
волоконно-оптический кабель 231, 250
восстановление ключей 869
восходящий порт 423
вредоносная программа 837
временное мультиплексирование 233, 276, 278
асинхронный режим 278
синхронный режим 278
статистическое 280
время
буферизации 97
жизни
записи 527
маршрута 573, 579
пакета 516, 527, 547
коммутации пакета 166
конвергенции 573
наработки на отказ 179
оборота 176, 366, 570
ожидания пакета в очереди 166
пакетизации 99
передачи
данных в канал 165
сообщения 97
распространения сигнала 98, 165
реакции сети 176
сериализации 165
удержания токена 372
Всемирная паутина 801
вторжение 888
вторичный задающий генератор 315
входная очередь 88
входной буфер 88, 100
выборка случайной величины 169
выделенный канал 662
выделенный сервер 49
выравнивание
заголовка пакета 517
отрицательное 323, 345
положительное 323, 345
высокоуплотненное волновое мультиплексирование 335
высокоуровневое управление линией связи 690
выходная очередь 88

Г

гарантированная доставка 613
гармоника
основная 259
понятие 233
генератор
вторичный 315
задающий 315
первичный 314
эталонный 314
геостационарная орбита 298
геостационарный спутник 298

гиперссылка 802
гипертекстовая информационная служба 34
гипертекстовая страница 802
гистограмма распределения 169
главное устройство 390
глобальная метка потока 64
глобальная сеть 28, 139, 232
глобальная система навигации 300
глобальный агрегируемый уникальный адрес 642
глобальный адрес 120
ГЛОНАСС 300
горизонтальная подсистема 252
горизонтальный контроль по паритету 274
городская сеть 34, 139
Гроша закон 28
группирование MAC-адресов 469
групповое вещание 621
  из конкретного источника 626
  из любого источника 626
групповой адрес 59, 360, 490, 491, 528

Д

дайджест 855
дайджест-функция 855
двоичная фазовая манипуляция 258
двоичная частотная манипуляция 258
двоичный код 52
двунаправленное обнаружение ошибок продвижения 717
двунаправленный виртуальный канал 673
двухточечная цепь 336
двухточечный протокол туннелирования 772
деградация системы 180
дейтаграмма 89, 116, 485
дейтаграммная передача 89
дейтаграммная сеть 141
дейтаграммный протокол 484
декомпозиция
  иерархическая 110
  понятие 109
демилитаризованная зона 541
демультиплексирование 69, 554
демультиплексор 70, 233
депонирование ключей 870
дерево 58
  кратчайшего пути 637
  разделяемое 631
  реверсивного кратчайшего пути 634
  с вершиной в источнике 631
  точки встречи 636
дескриптор потока 609
дескрэмблер 267, 270
десятичная упаковка 272
децибел 235
дешифрирование 848
джиттер 177
диапазон
  амплитудной модуляции 231
  инфракрасный 288
  микроволновый 231, 287
  очень высоких частот 231
  ультравысоких частот 231
  широковещательного радио 231
Дийкстры алгоритм 583
динамическая запись 409, 500
динамическая маршрутизация 573
динамическая страница 807
динамическая фрагментация 547
динамический номер порта 555
динамическое распределение кадров 462
диод
  лазерный 251
  светоизлучающий 251
дискретизация 262
  по времени 79, 261
  по значениям 79, 261
дискретная модуляция 261
диспетчер каналов 393
дистанционно-векторный алгоритм 574
дистанционно-векторный протокол маршрутизации 632
дифракционная структура 339
дифракционная фазовая решетка 339
дифракция 290
дифференцированное обслуживание 515, 603
диффузный передатчик 296
длина
  заголовка 515
  пульсации 433
долговременное соединение 805
долговременные характеристики сети 164
доля потерянных пакетов 179
домен
  группового вещания 630
  имен 504
  коллизий 418
  широковещательного трафика 468
доменная система имен 503
доменное имя 488, 503, 504
дополнительная величина пульсации 676
доставка с максимальными усилиями 91
достоверность передачи данных 242
доступ
  избирательный 859
  коллективный 362
  кочевой 377
  маркерный 356
  по требованию 427
  приоритетный 427
  резидентный 377
  случайный 356
  терминальный 767
доступность 179, 830

драйвер
переферийного устройства 41
сетевой интерфейсной карты 41
древовидная топология 141
дробный канал 313
дуплексная связь
с временным разделением 281
с частотным разделением 281
дуплексный канал 55
дуплексный режим коммутатора 418

Е

емкость канала связи 54

З

заголовок
аутентификации 645, 892
вставка 705
кадра 395
маршрутизации 645
мультиплексной секции 321
основной 645
пакета 85
пути 318
регенераторной секции 321
системы безопасности 645
следующий 645
тракта 323
фрагментации 645
задержка
доставки пакета 170
квантиль 171
коэффициент вариации 171
медиана 170
пакетизации 679
передачи кадра 424
процентиль 171
среднее значение 170
стандартное отклонение 170
закон Гроша 28
закрытый ключ 851
замена метки 675
замораживание изменений 582
запись
динамическая 409, 500
статическая 409, 500
запрещенный код 268
запрос
на резервирование ресурсов 609
понятие 81
затопление сети 409
затухание
погонное 236
понятие 235
защита
1:1 325
защита (*продолжение*)
1+1 325
1:N 326
антивирусная 872
карт 326
линии 722
мультиплексной секции 327
пути 723
сетевого соединения 328
узла 723
защитное переключение оборудования 326
защищенность кабеля 241
защищенный канал 889
защищенный протокол IP 646
звезда 57
звездообразная топология 57, 141
звено 229
значение проверки целостности 897

И

идентификатор
виртуального канала 94
запроса 596
интерфейса 643
коммутатора 450
организационно уникальный 360
пакета 516, 547
порта 451
соединения 94, 674
иерархическая адресация 60
иерархическая декомпозиция 110
иерархическая звезда 58
иерархия скоростей 311
избирательный доступ 859
избыточный код 268
измерение
активное 171
пассивное 173
изотропная антенна 287
изохронное приложение 189
импульсно-кодовая модуляция 262, 311
импульсный набор 770
импульсный способ кодирования 52
имя
DNS-имя 488
доменное 488
краткое 504
относительное 504
полное 504
плоское 502
символьное 488
индекс параметров безопасности 896
индивидуальный адрес 360, 490
индивидуальный клиент 147
инжиниринг
социальный 832
трафика 216, 718

инкапсуляция 670
интегрированная система управления 821
интегрированное обслуживание 603, 678, 775
интегрируемость сети 182
интеллектуальная сеть 36
интенсивность
  битовых ошибок 242
  отказов 179
интерактивное приложение 189
интерактивные услуги 147
интерактивный голосовой ответ 813
интервал
  hello 451
  битовый 365, 429
  межпакетный 363
  отсрочки 365, 429
Интернет 35, 156
интерфейс
  доступа к гигабитной среде 436
  логический 41
  межуровневый 110
  начальный 778
  независимый от среды 429
  ненумерованный 693
  одноранговый 112
  основной 778
  понятие 40
  прикладной программный 115
  связи между частными сетями 681
  сетевой 59
  сеть-сеть 749
  услуг 110
  физический 41
  шлюзовой 808
интерфейсная карта 41
интранет 684
инфокоммуникационная сеть 36, 132
информационная безопасность 830, 846
информационные услуги 131, 147, 154
информационный поток 63, 485
информационный центр 143, 144
инфракрасные волны 288
инфракрасный диапазон 288
инфраструктура с открытыми ключами 869
истечение времени жизни маршрута 579

К

кабель 41
  волоконно-оптический 231, 250
  категории 1 248
  категории 2 248
  категории 3 248
  категории 4 248
  категории 5 248
  категории 6 236, 249
  категории 7 249
  коаксиальный 231, 249
кабель (*продолжение*)
  медный 231
  многомодовый 250
  неэкранированный 248
  одномодовый 250
  симметричный 247
  телевизионный 250
кабельная линия связи 231
кабельный модем 765
кадр 486
  STM-N 317
  положительной квитанции 381
  помеченный 471
  понятие 116
  продвижение 409
  состав 116
канал 276
  арендуемый 662
  асинхронный 392
  виртуальный 94, 680
  выделенный 662
  доступа 101
  дробный 313
  дуплексный 55
  коммутируемый 680
  не ориентированный на соединение 392
  оптический 344
  ориентированный на соединение 392
  полудуплексный 55
  понятие 78, 229
  постоянный 680
  присоединения 737
  связи 41
  симплексный 55
  синхронный 392
  составной 80, 229
  спектральный 334
  типа
    B 777
    D 778
    H 778
  тональной частоты 258
  уплотненный 277
  элементарный 78, 79, 262
канальный уровень 116
качество обслуживания 36
квадратурная амплитудная модуляция 259
квадратурная фазовая манипуляция 258
квантиль 171
квитанция 54
квитирование 564, 861
Кеплер Йоханес 298
Керкхоффа правило 848
класс
  адресов 489
    A 490
    B 490

класс (*продолжение*)
C 490
D 490
E 490
транспортного сервиса 121
трафика 611
услуги 704
эквивалентности продвижения 703
классификация
компьютерных сетей 139
критерии 139
трафика 197, 203
клиент
индивидуальный 147
корпоративный 147, 148
массовый 148
понятие 45
почтовый 795
клиентская операционная система 49
ключ
аппаратный 862
закрытый 851
открытый 849, 851
программный 863
секретный 848, 864
коаксиальный кабель 231, 249
толстый 249
тонкий 250
код
2D1Q 268
4B/5B 268, 429, 430
8B/6T 269, 429
AMI 266
B8ZS 270
HDB3 270
NRZ 264
биполярный импульсный 267
двоичный 52
доступа 394
запрещенный 268
избыточный 268
манчестерский 267
переменной длины 273
решетчатый 259, 276
самосинхронизирующийся 264
сверточный 276
Хемминга 275
кодирование 256
без возвращения к нулю 264
биполярное с альтернативной инверсией 266
импульсный способ 52
линейное 244
относительное 273
понятие 52
потенциальный способ 52
статистическое 273
кодирование (*продолжение*)
физическое 244
Хафмана 273
кодовый бит 559
коллективный доступ 362
коллизия 364
обнаружение 364
предотвращение 376
распознавание 366
кольцевая топология 57, 141, 338
кольцо 57
SDH 324
плоское 324
комбинированное обслуживание 202
комбинированный коммутатор 441
комитет производителей компактного оборудования 443
коммуникационное облако 47
коммутатор 233
2-го уровня 655
3-го уровня 469, 655
комбинированный 441
корневой 450
назначенный 451
неблокирующий 419
пограничный 748
понятие 67, 68
программный 815
с общей шиной 440
стековый 445
с фиксированным количеством портов 442
фотонный 340
коммутационная матрица 414, 438
коммутационная сеть 68
коммутация
интерфейсов 67
каналов 36, 73, 123
многопротокольная 698
на лету 415
на основе тегов 700
напролет 415
пакетов 36, 73, 85, 123
по меткам 702
понятие 62, 68
коммутируемый виртуальный канал 680
коммутируемый модем 764
коммутирующий блок 88, 652
коммутирующий по меткам маршрутизатор 698, 702
компрессия
данных 272
трафика 187
компьютер-бастион 882
компьютерная сеть 25, 44
компьютерный трафик 84
конвейерная передача 805
конвергенция 573

конвертор интерфейса Gigabit Ethernet 443
кондиционирование трафика 186, 202
конечная точка обслуживания 743
конкатенация
  виртуальная 331
  смежная 331
конкурентное окно 381
конкурс 291
контейнер виртуальный 318
контент 159, 878
контролируемый период 383
контроллер 42
контроль
  допуска в сеть 210
  доступа 203
  по паритету 274
    вертикальный 274
    горизонтальный 274
  потока 206
  расходования ресурсов 164
  циклический избыточный 275
контрольная последовательность кадра 168, 274, 361
контрольная сумма 54, 105, 274
  заголовка 516
  пакета 85
контрольная точка 781
конфигурационный параметр 508
конфигурирование 508
конфиденциальность 830
концевик 85
концентратор 232, 369
  понятие 57
  удаленного доступа 765
корневой коммутатор 450
корневой порт 451
корпоративная сеть 142, 145, 151, 154
корпоративный клиент 147, 148
корпоративный маршрутизатор 653
корпоративный сетевой экран 876
коррекция ошибок 275
кочевой доступ 377
коэффициент
  вариации 171
  использования 193
  пульсации трафика 178
  расширения 305
кража бита 312
кратковременное соединение 805
краткое доменное имя 504
краткосрочные характеристики сети 164
кратчайший маршрут 216
кредит 208
криптосистема
  ассимметричная 851
  понятие 848
криптосистема (*продолжение*)
  раскрытие 848
  симметричная 849
криптостойкость 848
критерий
  выбора маршрута 515
  классификации 139
кросс-коннектор 320
кэширование данных 187

Л

лавинная маршрутизация 572
лазерный диод 251
линейное кодирование 244
линия
  доступа 472
  связи 52, 229
    аналоговая 233
    воздушная 230
    кабельная 231
    качество 33
    проводная 230
    радиорелейная 292
    создание 41
    цифровая 233
лицензия 149, 291
логический интерфейс 41
логический порт 462
логическое соединение 91, 561
локализация адресов 546
локальная метка потока 64
локальная сеть 31, 139
локальная служба распределения 790
локальная таблица коммутации 82
локальное приложение 49
локальный адрес 486
локальный оператор 149
локальный поставщик услуг 159
локальный признак потока 82
локальный способ назначения адреса 360
лотерея 291
лямбда 333

М

магистраль 143
магистральная сеть 142–144
магистральный маршрутизатор 651
магистральный порт 423
магистральный поставщик услуг 159
магнитная связь 239
максимальная емкость адресной таблицы 425
максимальная скорость передачи 208
маловысотная орбита 298
мандатный подход 859

манипуляция
амплитудная 257
фазовая 257
двоичная 258
квадратурная 258
частотная 257
двоичная 258
многоуровневая 258
четырехуровневая 258
манчестерский код 267
маркер доступа 372
маркерный доступ 356
марковское распределение 192
маршрут
временный 528
кратчайший 216
определение 120
понятие 62
постоянный 528
по умолчанию 521
специфический 520, 528
статический 528
маршрутизатор 68, 119
волн 340
доступа 653
коммутирующий по меткам 698, 702
корпоративный 653
локальной сети 655
магистральный 651
оператора связи 653
пограничный 653, 702
по умолчанию 521
программный 523, 655
регионального отделения 654
удаленного офиса 655
маршрутизация 68
адаптивная 573
динамическая 573
лавинная 572
от источника 572
статическая 573
маршрутизируемый протокол 121
маршрутизирующий протокол 121
маска 489, 533
двоичная запись 489
понятие 489
массовый клиент 148
масштабируемость сети 180, 482
матрица коммутационная 414, 438
медленное расширение спектра 303
медный кабель 231
медный неэкранированный кабель 248
межпакетный интервал 363
межсетевое взаимодействие 118
межсетевой протокол 484, 514
межсетевой экран 876
межсимвольная интерференция 290
межуровневый интерфейс 110
метасимвол 601
метка
замена 675
назначение 675
потока 94
глобальная 64
локальная 64
Меткалф Роберт 354
метод
инжиниринга трафика 186, 216, 217
кондиционирования трафика 186
контроля перегрузок 164
маркерного доступа 356
обеспечения качества обслуживания 185
обратной связи 186
опроса 376
предотвращения перегрузок 164
простоя источника 564
скользящего окна 565
случайного доступа 356
метрика 450
понятие 65
производительности сети 168, 174
механизм
кондиционирования трафика 202
обратной связи 421
предотвращения перегрузки 205
управления перегрузкой 205
микроволновая система 287
микроволновый диапазон 287
микроэлектронная механическая система 341
миниатюрный апертурный терминал 299
мини-компьютер 30
минимальная таблица маршрутизации 527
многоканальная служба распределения 790
многоканальный протокол PPP 690
многолучевое замирание 290
многолучевое распространение сигнала 290
многомодовое оптическое волокно 250
многомодовый кабель 250
многопротокольная коммутация с помощью меток 698
многотерминальная операционная система 29
многотерминальная система разделения времени 26
многоуровневая частотная манипуляция 258
многоуровневый подход 109
множественный доступ с кодовым разделением 276, 306
множественный протокол покрывающего дерева 474
мобильная беспроводная связь 285

мобильная компьютерная сеть 286
мобильная телефония 285
мода 250
модель взаимодействия открытых систем 108, 113
модем 232, 258
  кабельный 765
  коммутируемый 764
  телефонный 764
модемная терминальная станция 788
модуляция 53, 256
  амплитудная 231, 257–259
  дискретная 261
  импульсно-кодовая 262, 311
  квадратурная 259
  понятие 245
  с несколькими поднесущими 302
  фазовая 258
  частотная 231, 257, 258
мост
  локальной сети 403
  понятие 403
  провайдера 746
  прозрачный 406, 407
  с маршрутизацией от источника 407
  транслирующий 407
мощность опорная 237
мультиплексирование 69, 555
  волновое 276–278, 310, 333–335
  временное 233, 276, 278
  высокоуплотненное 335
  ортогональное 302
  пространственное 389
  уплотненное 278, 310, 333
  частотное 233, 276, 302
мультиплексная секция 321
мультиплексор 70, 233, 319
  ввода-вывода 320, 338
  доступа 765, 786
  оптический 338
  терминальный 320
мультипрограммная операционная система 102
мультисервисная сеть 35
мэйнфрейм 26

Н

набор
  импульсный 770
  тоновый 770
  услуг
    базовый 378
    расширенный 379
наведенный сигнал 240
наводка 235
  объединенная 240
  перекрестная 240
  понятие 240
надежность транспортных услуг 164
назначение
  динамических адресов 509
  статических адресов
    автоматическое 509
    ручное 508
назначенный коммутатор 451
назначенный порт 451
Найквиста–Котельникова теорема 262
Найквиста теория 262
Найквиста формула 228, 246
наложенная сеть 141, 230
направленная антенна 287
национальный оператор 149
начальное число 303
начальный интерфейс 778
неблокирующий коммутатор 419
недогруженная сеть 186
недогруженный режим 222
независимое поведение маршрутизаторов 612
независимый от среды интерфейс 429
незащищенное соединение 762
ненаправленная антенна 287
ненаправленная среда 287
ненумерованный интерфейс 693
необратимая функция 855
неопределенный адрес 491
неполносвязная топология 57
нерекурсивная процедура разрешения имени 506
несущая частота 245, 362
несущий сигнал 244
неумышленная угроза 831
неэкранированная витая пара 231, 247, 248
неэкранированный кабель 248
низкоорбитальный спутник 300
номер
  версии протокола 515
  порта
    динамический 555
    назначенный 555
    стандартный 555
    хорошо известный 555
  сети 487, 488
  узла в сети 487, 488
номинальная скорость протокола 370

О

обеспечение информационной безопасности 846
область сети 585
обнаружение
  коллизии 364
  ошибок 274
  случайное раннее 607
обновление триггерное 582

оборудование
коммуникационное 682
кроссовое 682
терминальное 144, 781
обработка ошибок 822
обратная доставка 316
обратная зона 507
обратная петля 491
обратная связь 186, 421
обслуживание
взвешенное 200, 202
дифференцированное 515, 603
интегрированное 603, 678, 775
комбинированное 202
приоритетное 197
справедливое 202
общая длина пакета 516
общая среда передачи данных 354
общая шина 58, 103, 440
общедоступный домен Интернета 765
общий шлюзовой интерфейс 808
объединение подсетей 545
объединенная наводка 240
объем пульсации 606
объявление
о расстоянии 574
о состоянии связей сети 583
ограниченная широковещательная рассылка 491
ограниченный широковещательный адрес 491
ограничитель начала кадра 363
одномодовое оптическое волокно 250
одномодовый кабель 250
однонаправленный виртуальный канал 673
однопрограммная операционная система 102
одноразовый пароль 862
одноранговая операционная система 49
одноранговый интерфейс 112
односторонняя задержка пакетов 174
односторонняя функция 855
окно
конкурентное 381
приема 568
прозрачности 237
скользящее 565
оконечное оборудование данных 232
оператор
локальный 149
национальный 149
операторов 149
региональный 149
связи 145
транснациональный 150
операционная система
клиентская 49
компьютера 47
многотерминальная 29
операционная система (*продолжение*)
мультипрограммная 102
однопрограммная 102
одноранговая 49
серверная 49
сетевая 29, 47
опорная мощность 237
оптическая транспортная сеть 310, 342
оптический канал 344
оптический кросс-коннектор 338, 340
оптический мультиплексор 338, 339
оптоэлектронный кросс-коннектор 340
орбита
геостационарная 298
маловысотная 298
средневысотная 298
организационно уникальный идентификатор 360
ортогональное частотное мультиплексирование 302
основная гармоника 259
основной заголовок 645
основной интерфейс 778
особый IP-адрес 527
отказ
в обслуживании 832
в установлении соединения 82
отказоустойчивость 180
открытая система 124
открытая спецификация 124
открытый ключ 849, 851
относительное доменное имя 504
относительное кодирование 273
относительный коэффициент использования 201
относительный уровень мощности 237
отображение нагрузки
асинхронное 345
синхронное 345
отрицательное выравнивание 323, 345
офисный телефонный коммутатор 144
очередь
FIFO 192, 197
взвешенная 200
входная 88
выходная 88
повторной передачи 570
приоритетная 197
ошибка переполнения буфера 841

П

пакет 116, 119, 485
пакетная сеть 689
пакетный коммутатор 88
пакетный метод коммутации 36
память
буферная 88

память (*продолжение*)
  многовходовая 441
  разделяемая 441
параболическая антенна 287
параметры
  логического соединения 91
  получателя 645
  специальные 645
пароль
  одноразовый 862
  применение 858
пассивное измерение 173
первичная сеть 141, 230, 310
первичный эталонный генератор 314
перегрузка
  контроль 164
  предотвращение 164, 205
  признак 208
  управление 205
передача
  голоса 29
  дейтаграммная 89
  конвейерная 805
  последовательная 805
  с простоями 805
  с установлением
    виртуального канала 93
    логического соединения 91
  эстафетная 295
перекрестная наводка
  на ближнем конце 240
  на дальнем конце 240
переменная битовая скорость 188
переполнение буфера 841
перераспределение 586
период
  контролируемый 383
  пульсации 178
персональная сеть 351, 389
персональный компьютер 32
персональный сетевой экран 876
петля 411
пиковая скорость передачи данных 178
пикосеть 390
пилотный сигнал 307
планирование
  расходования ресурсов 164
  сети 182
плезиохронная цифровая иерархия 310
плоская адресация 60
плоское имя 502
плоское кольцо 324
плотный режим 630
повторитель 232
погонное затухание 236
пограничный коммутатор 748
пограничный маршрутизатор 653, 702
пограничный шлюзовой протокол 588
поддомен 504
подпись
  цифровая 870
  электронная 870
подпоток 63
подсеть 493
подсистема
  вертикальная 252
  горизонтальная 252
  кампуса 252
подуровень управления 421
подчиненное устройство 390
покрывающее дерево 412, 449
поле
  данных 361, 394
  источника 527
  контрольной последовательности кадра 361
  следующего заголовка 645
полезная пропускная способность протокола 371
политика информационной безопасности 847
полное доменное имя 504
полносвязная топология 56, 141
полностью оптическая сеть 336
полностью оптический кросс-коннектор 340
положительная квитанция 381
положительное выравнивание 323, 345
полоса пропускания 54, 242
полудуплексный канал 55
полудуплексный режим коммутатора 418
полупроводниковый лазер 251
пользовательский слой 135
пользовательский фильтр 411, 465
помеха
  внешняя 235
  внутренняя 235
помехоустойчивость 239
помеченный кадр 471
порог чувствительности приемника 239
порт 41
  TCP-порт 555
  UDP-порт 555
  агрегатный 319
  альтернативный 457
  восходящий 423
  доступа 472
  корневой 451
  логический 462
  магистральный 423
  назначенный 451
  приложения 555

порт (*продолжение*)
резервный 457
с разделением каналов 693
трибутарный 319
физический 462
порядковый номер запроса 596
последовательная передача 805
последовательность
Баркера 305
псевдослучайной перестройки частоты 303
расширяющая 305
поставщик услуг
биллинговых 160
Интернета 157, 159
локальный 159
магистральный 159
по доставке контента 159
по поддержке приложений 160
региональный 159
хостинга 159
постоянная битовая скорость 187
постоянный виртуальный канал 673, 680
построение Интернета будущего 829
потенциальный код
2D1Q 268
NRZ 264
без возвращения к нулю 264
с инверсией при единице 266
потенциальный способ кодирования 52
поток
агрегированный 615
байтов 558
данных 63, 485
информационный 63, 485
контроль 206
потоковый трафик 187
почтовый клиент 795
почтовый сервер 795
пошаговая спецификация 613
преамбула 363, 366
предложенная нагрузка 54, 166
предотвращение
коллизий 376
перегрузки 205
преобразователь
аналого-цифровой 261
цифро-аналоговый 261
префикс 495, 545
адреса 495
формата 642
привратник 811
признак
непосредственно подключенной сети 526
перегрузки 208
прикладной программный интерфейс 115
прикладной уровень 123, 483
приложение
асинхронное 189
изохронное 189
локальное 49
сверхчувствительное к задержкам 189
сетевое
распределенное 49
централизованное 49
синхронное 189
с потоковым трафиком 187
с пульсирующим трафиком 188
устойчивое к потере данных 189
чувствительное к потере данных 189
приоритет
пакета 515
понятие 197
приоритетная очередь 197
приоритетное обслуживание 197
приоритетный доступ по требованию 427
проблема последней мили 759
провайдер 157
проверка непрерывности соединения 743
проводная сеть 140
проводная среда 230, 287
проводное абонентское окончание 790
программа
вредоносная 837
троянская 838
шпионская 844
программное обеспечение стека TCP/IP 527
программный ключ 863
программный коммутатор 815
программный маршрутизатор 523, 655
продвижение
кадра 409
по реверсивному пути 632
прозрачный мост 406, 407
производительность
коммутатора 425
транспортных услуг 164
прокси-сервер
понятие 883
прикладного уровня 886
уровня соединений 886
промежуточная аппаратура 232
промежуточная точка обслуживания 743
пропускная способность 54, 243
простая фильтрация 880
простой источника 564
простой протокол
передачи почты 483, 796
управления сетью 135
пространственное мультиплексирование 389
протокол
IPSec 646
Proxy-ARP 501

протокол (*продолжение*)
адаптации 393
аутентификации 861
по квитированию вызова 690
по паролю 690
беспроводных приложений 804
верхнего уровня 516
взаимодействия приложений 43
группового управления в Интернете 627
двунаправленного обнаружения ошибок продвижения 717
двухточечной связи 690
дейтаграммный 484
динамического конфигурирования хостов 508
доступа
к линии связи для модемов 775
к электронной почте 800
загрузки 637
инициирования сеанса 811
интеллектуальной сети 816
как логический интерфейс 41
коррекции ошибок 774
маршрутизации 121, 528, 553
группового вещания 627
дистанционно-векторный 632
маршрутизируемый 121
маршрутной информации 575
межсетевой 484, 514
межсетевых управляющих сообщений 484, 591
общей управляющей информации 826
ориентированный
на передачу 796
на прием 796
передачи
гипертекста 483, 805
почты 483, 796
файлов 483, 819
покрывающего дерева
быстрый 449
классический 449
множественный 474
пользовательских дейтаграмм 483, 554
понятие 112
почтового отделения 800
разрешения адресов 61, 487, 497, 651
распределения меток 701, 709
реального времени 809
резервирования ресурсов 604, 609
сжатия синхронных потоков данных 775
сигнализации 681
сигнальный 312, 609, 701, 769
туннелирования 772
управления
агрегированием линий связи 464
линией связи 690
управления (*продолжение*)
передачей 483, 554
сетью 135, 690
шлюзовой
внешний 588
внутренний 588
пограничный 588
эмуляции терминала 483
протокольная единица данных 116, 451
профилирование 203
профиль 393
процедура
разрешения имени
нерекурсивная 506
рекурсивная 506
установления соединения 81
процентиль 171
процессор
пакетов Ethernet 414
цифровой обработки сигнала 435
прямая коррекция ошибок 275
прямое последовательное расширение спектра 305
псевдоканал 733
пуассоновское распределение 192
пул адресов 510
пульсация 433
пульсирующий трафик 84
путь
виртуальный 681
коммутации по меткам 702
ПЭГ 314

Р

радиодиапазон 287
радиоканал 231
радиорелейная линия связи 292
радиосеть 147
разброс задержки 177
разделение
времени 70
каналов 693
на подсети 545
ресурсов 40
частотное 70
разделяемая многовходовая память 441
разделяемая среда 71
разделяемое дерево 631
размер окна 565
разрешение адреса 61
разряженный режим 630
распознавание коллизий 366
распределение
динамическое 462
марковское 192
пуассоновское 192
статическое 463

распределенная атака 832
распределенная система 379
распределенное приложение 49
распределенный режим 380
распределитель 764
рассредоточенная сеть 392
расстояние Хемминга 275
рассылка
  ограниченная 491
  широковещательная 491
расширение
  поля данных кадра 433
  спектра
    быстрое 303
    медленное 303
    прямое последовательное 305
    скачкообразной перестройкой частоты 302
расширенный интерфейс 436
расширенный набор услуг 379
расширенный спектр 302
расширенный список доступа 602
расширяемость сети 180
расширяющая последовательность 305
расщепление горизонта 581, 741
реальная частная сеть 682
реверсивный протокол разрешения адресов 501
регенераторная секция 321
регенератор сигнала 232, 320
региональный оператор 149
региональный поставщик услуг 159
режим
  аутентификации 691
  включения 626
  дуплексный 418
  исключения 626
  неблокирующий 419
  недогруженный 222
  неразборчивого захвата 361, 407
  передачи
    асинхронный 280, 678
    синхронный 280
  перераспределения 586
  плотный 630
  полудуплексный 418
  пульсаций 433
  разряженный 630
  распределенный 380
  свертывания колец 374
  терминального доступа 767
  транспортный 894
  туннельный 894
  удаленного узла 765
  удаленного управления 767
  централизованный 380
резервирование
  пропускной способности 210
  ресурсов 209, 609
резервная связь 412
резервное копирование 846
резервный порт 457
резидентный доступ 377
рекомендуемый стандарт 232
рекурсивная процедура разрешения имени 506
ресурсы
  контроль расходования 164
  планирование расходования 164
  разделение 40
ретрансляционный участок 519
решетчатый код 259, 276
риск 831
ручное назначение статических адресов 508

С

самовосстанавливающаяся сеть 325
самосинхронизирующийся код 264
сверточный код 276
свертывание колец 374
сверхвысокая частота 287
сверхнизкая частота 287
световод 250
светодиод 251
светоизлучающий диод 251
свободный TE-туннель 718
связной агент 511
связь
  ближнего радиуса действия 398
  магнитная 239
  наземная 231
  резервная 412
  спутниковая 231
  электрическая 239
сеанс
  передачи данных 820
  управляющий 820
сеансовый уровень 122
сегмент 116, 449, 485, 558
секретный ключ 848, 864
секция
  мультиплексная 321
  регенераторная 321
сервер
  выделенный 49
  имен 61
  маршрутов 573
  понятие 46
  почтовый 795
  сетевой 32
  удаленного доступа 765
серверная операционная система 49
сервис 613
сертификат 865
сетевая интерфейсная карта 41
сетевая операционная система 29, 47

сетевая служба 46
сетевая технология 31
сетевое окончание 777
сетевой адаптер 41
сетевой адрес 120, 487
  выходного интерфейса 520
  следующего маршрутизатора 520
сетевой интерфейс 59, 732
сетевой монитор 517
сетевой протокол Microcom 775
сетевой сервер 32
сетевой уровень 118, 484
сетевой червь 838
сетевой экран
  корпоративный 876
  персональный 876
  понятие 876
  прикладного уровня 880
  сеансового уровня 880
  сетевого уровня 880
  с фильтрацией пакетов 880
сеть
  агрегирования трафика 142
  беспроводная 140, 375
  виртуальная 467, 662, 682
  внешняя 684
  внутренняя 684
  глобальная 28, 139, 232
  городская 34, 139
  дейтаграммная 141
  демилитаризованной зоны 882
  доступа 142, 143
  затопление 409
  здания 153
  интегрируемость 182
  интеллектуальная 36
  инфокоммуникационная 36, 132
  кампуса 153
  коммутационная 68
  компьютерная 25, 44, 286
  корпоративная 142, 145, 151, 154
  локальная 31, 139, 467
  магистральная 142—144
  масштаба предприятия 154
  масштабируемость 180
  мегаполиса 34, 139
  мобильная 286
  мультисервисная 35
  на базе
    виртуальных каналов 141
    логических соединений 141
  наложенная 141, 230
  недогруженная 186
  оператора связи 141, 145
  оптическая 310, 316, 342
  отдела 151
  первичная 141, 230, 310
сеть (*продолжение*)
  передачи данных 25, 35
  периметра 882
  персональная 351, 389
  планирование 182
  полностью оптическая 336
  проводная 140
  рабочей группы 152
  радио 147
  рассредоточенная 392
  расширяемость 180
  самовосстанавливающаяся 325
  с базовым набором услуг 378
  с избыточной пропускной способностью 186
  с интегрированным обслуживанием 35
  синхронная 316
  с коммутацией
    каналов 140
    пакетов 140
  совместимость 182
  составная 118
  с расширенным набором услуг 379
  телевизионная 147
  телефонная 28, 230
  транспортная 310, 342
  управляемость 181
  частная 662, 682
сжатие 272
сигнал
  наведенный 240
  несущий 244
  пилотный 307
  стартовый 42
  стоповый 42
сигнальная система 7 771
сигнальный протокол 312, 609, 701, 769
сигнатура вируса 873
символьное имя 488
символьное подавление 273
символьный адрес 59
симметричная криптосистема 849
симметричный кабель 247
симплексный канал 55
синхронизация передатчика и приемника 53, 263
синхронная оптическая сеть 316
синхронная цифровая иерархия 310
синхронное отображение нагрузки 345
синхронное приложение 189
синхронный канал 392
синхронный режим
  временного мультиплексирования 278
  передачи 280
система
  Т-каналов 311
  автономная 587

система (*продолжение*)
аутентификации 862
беспроводных абонентских окончаний 288
видимого света 288
дифференцированного обслуживания 603
доменных имен 488, 505
имен 503
интегрированного обслуживания 603
инфракрасных волн 288
кабельная 252
микроволновая 287
микроэлектронная механическая 341
многотерминальная 26
навигации глобальная 300
обнаружения вторжений 888
открытая 124
пакетной обработки 26
разделения времени 26
распределенная 379
сигнальная 771
терминальная 765
управления
сетью 821
системой 822
шифрования 854
скользящее окно 565
скорость
OLE_LINK7передачи данных 177
битовая 54
передачи данных 54
пиковая 178
предложенной нагрузки 54
продвижения 424
протокола номинальная 370
согласованная 675
средняя 178
фильтрации 424
чиповая 305
скремблирование 269
скрытый терминал 376
скрэмблер 267
слой
защищенных сокетов 122, 891
менеджмента 135
пользовательский 135
управления 135
слот трибутарный 346
служба
безопасности 47
каталогов 47
мониторинга сети 47
печати 46
распределения
локальная 790
многоканальная 790
распределенной системы 379
резервного копирования и архивирования 47
сетевая 46
справочная 47
файловая 46
случайное раннее обнаружение 607
случайный доступ 356
смежная конкатенация 331
смешанная топология 58, 141
смещение фрагмента 516
сниффер 844
совет по архитектуре Интернета 126
совместимость сети 182
согласованная величина пульсации 676
согласованная скорость передачи данных 675
соглашение об уровне обслуживания 165, 822
соединение
виртуальное 681
долговременное 805
кратковременное 805
логическое 91, 561
незащищенное 762
отказ в установлении 82
установление 81
сокет 556
сообщение 43, 97
PATH 609
RESV 609
начального адреса 781
понятие 116, 123
проверки непрерывности соединения 743
электронное 795
сообщество Интернета 126
сопротивление
активное 239
волновое 239
составная сеть 118
составной канал 80, 229
состояние защищенности 830
сота 294
сохранение с продвижением 88
социальный инжиниринг 832
спам 845
спектр
кода 271
расширенный 302
сигнала 233, 259, 263
спектральное разложение сигнала 233
спектральный канал 334
специальные параметры 645
спецификация
запроса приемника 609
открытая 124
пошаговая 613
трафика источника 609
фильтра 609

специфический маршрут 520, 528
список
доступа 465, 600
расширенный 602
стандартный 600
контроля доступа 203
справедливое обслуживание 202
спутник
геостационарный 298
низкоорбитальный 300
среднеорбитальный 300
спутниковая связь 231
среда
беспроводная 230, 287
ненаправленная 287
проводная 230, 287
разделяемая 71
физическая 230
средневысотная орбита 298
среднеорбитальный спутник 300
среднесрочные характеристики сети 164
средняя скорость
передачи данных 178
поступления маркеров 606
срок аренды 509
стадия
обучения 454
продвижения 454
прослушивания 454
стандарт
комитетов и объединений 125
международный 125
межсетевого взаимодействия 357
межуровневого взаимодействия 816
на кабельные системы 252
национальный 125
рекомендуемый 232
сжатия данных 775
фирменный 125
стандартная сетевая технология 32
стандартная топология физических связей 354
стандартный назначенный номер порта 555
стандартный список доступа 600
стартовый сигнал 42
статистическая оценка 170
статистическое временное мультиплексирование 280
статистическое кодирование 273
статическая запись 409, 500
статическая маршрутизация 573
статическая страница 807
статический маршрут 528
статическое распределение кадров 463
стек
TCP/IP 130, 483
стек (*продолжение*)
коммуникационных протоколов 112
меток 706
протоколов SDH 320
стековый коммутатор 445
стоповый сигнал 42
страница
гипертекстовая 802
динамическая 807
статическая 807
строгая аутентификация 858
строгий TE-туннель 718
структурированная кабельная система 252
схема автопереговоров 430
сшивание путей 706

Т

таблица
адресная 407, 408
коммутации 66, 68, 90, 95
кросс-соединений 318
маршрутизации 68, 120, 519
минимальная 527
формирование 527
продвижения 408, 701
соединений 318
соответствия адресов 61
фильтрации 408
тайм-аут 579
доставки 122
квитанции 570
таймер отсрочки 381
тайм-слот 278
такт 245
тег
виртуальной локальной сети 471
языка разметки 802
телевизионная сеть 147
телевизионный кабель 250
телефон аналоговый 770
телефонная сеть 28, 230
телефонные услуги 146
телефонный модем 764
тема для обсуждения 126
темное волокно 667
теорема Найквиста—Котельникова 262
теория
автоматического управления 207
Найквиста 262
очередей 191
терминальная система 765
терминальное оборудование 144, 777, 781
терминальный адаптер 232, 783
терминальный доступ 767
терминальный мультиплексор 320
тест целостности соединения 370

техника расширенного спектра 302
технология
  бесклассовой междоменной маршрутизации 494, 544
  волнового мультиплексирования 334
  коммутации на основе тегов 700
  межсетевого взаимодействия 118
  сетевая 31
  цифровых сетей с интегрированным обслуживанием 35
тип сервиса 515
токен доступа 372
толстый коаксиальный кабель 249
тонкий коаксиальный кабель 250
тонкопленочный фильтр 339
тоновый набор 770
топология
  древовидная 58, 141
  звездообразная 57, 141
  кольцевая 57, 141, 338
  неполносвязная 57
  полносвязная 56, 141
  понятие 55
  смешанная 58, 141
  ячеистая 57, 324, 338
точка
  встречи 631
  доступа 294, 379
  классификации трафика 197
  контрольная 781
  обслуживания
    конечная 743
    промежуточная 743
  присутствия 150
  рандеву 631
традиционная технология NAT 616
транк 459, 472
транслирующий мост 407
трансляция сетевых адресов 616
  базовая 617
  двойная 621
  и портов 617
транснациональный оператор 150
транспондер 667
транспорт Ethernet операторского класса 730
транспортное средство 47
транспортные услуги 131, 147
  безопасность 164
  надежность 164
  производительность 164
транспортный блок оптического канала 344
транспортный режим 894
транспортный уровень 121, 483
транспортный шлюз 815
трафик 135
  инжиниринг 186, 216, 217
трафик (*продолжение*)
  классификация 197, 203
  компрессия 187
  компьютерный 84
  кондиционирование 186, 202
  межсегментный 406
  неравномерный 96
  потоковый 187
  профилирование 203
  пульсирующий 84
  формирование 204
  эластичный 189
трибутарный блок 318
трибутарный порт 319
трибутарный слот 346
триггерное обновление 582
троянская программа 838
туннелирование 670
туннельный режим 894

У

угроза
  внешняя 831
  неумышленная 831
  понятие 831
  умышленная 831
удаление метки на предпоследнем хопе 704
удаленное управление 767
удаленный доступ 759
удаленный узел 765
узкое место составного пути 179
указатель
  виртуального контейнера 318
  срочности 571
улучшенная скорость передачи данных 394
умышленная угроза 831
уникальный адрес 59
унифицированный указатель ресурса 803
уплотненное волновое мультиплексирование 278, 310, 333
уплотненный канал 277
управление
  активное 206
  безопасностью 822
  выравниванием 346
  доступом к среде 116, 355, 358
  конфигурацией сети и именованием 821
  логическим каналом 358, 393
  очередями 186, 206
  перегрузкой 205
управляемость сети 181
управляющий сеанс 820
уровень
  аудиоуровень 394
  базового диапазона частот 393
  Интернета 484

уровень (*продолжение*)
канальный 116
линии 321
мощности
абсолютный 237
относительный 237
представления 122
прикладной 123, 483
протокола адаптации 393
сеансовый 122
секции 321
сетевой 118, 484
сетевых интерфейсов 484
согласования 429
тракта 321
транспортный 121, 483
управления 394
физический 116
физических радиосигналов 393
фотонный 320
усеченный экспоненциальный алгоритм отсрочки 365
усилитель 232
ускоренная MPLS-коммутация 709
услуги
виртуальной частной локальной сети 739
интерактивные 147
информационные 131, 147, 154
компьютерных сетей 146
телефонные 146
транспортные 131, 147
широковещательные 147
установление логического соединения 91
устройство
главное 390
для подключения к цифровым каналам 232
подчиненное 390
сетевого окончания 783
физического уровня 429
учет работы сети 822

Ф

фазар 340
фазовая манипуляция 257
фазовая модуляция 258
файервол 876
файловая служба 46
файл-сервер 46
физическая среда передачи данных 230
физический интерфейс 41
физический порт 462
физический уровень 116
физическое кодирование 244
фиксированная беспроводная связь 285
фиксированная граница адреса 489
фильтр
пользовательский 411, 465
тонкопленочный 339
фильтрация 424
кадра 409
маршрутных объявлений 603
пользовательского трафика 600
понятие 600
простая 880
с учетом контекста 880
флаг пакета 516
формирование трафика 204
формула
Найквиста 228, 246
Фурье 235
Шеннона 228, 241, 246
фотонный коммутатор 340
фотонный уровень 320
фрагментация 547
фрейм 486
фронт 264
функция
необратимая 855
односторонняя 855
Фурье формула 235

Х

хаб 369
характеристики
задержек пакетов 169
качества обслуживания 185
сети
долговременные 164
краткосрочные 164
производительность 168
среднесрочные 164
Хафмана алгоритм 273
Хемминга расстояние 275
хоп 519
хорошо известный номер порта 555
хост 483
хэш-функция 855

Ц

ЦАП 261
целостность 830, 897
центр
информационный 143, 144
обмена трафиком 159
сертификации 865
управления сервисами 143, 144
централизованная справочная служба 47
централизованное сетевое приложение 49
централизованный режим 380

централизованный способ назначения адреса 360
центральный блок управления 652
центральный офис 150
цепь 324
  двухточечная 336
  с промежуточными подключениями 337
циклический избыточный контроль 275
цифро-аналоговый преобразователь 261
цифровая иерархия
  плезиохронная 310
  синхронная 310
цифровая линия связи 233
цифровая оболочка 342
цифровая обработка сигналов 281
цифровая подпись 870
цифровая сеть с интегрированным обслуживанием 775
цифровое абонентское окончание 777
цифровой кросс-коннектор 320
цифровой сертификат 864

Ч

частная линия Ethernet 733
частная сеть 148
  виртуальная 682
  реальная 682
частный адрес 493, 616
частота
  несущая 245, 362
  сверхвысокая 287
  сверхнизкая 287
частотная манипуляция 257
частотная модуляция 231, 257, 258
частотное мультиплексирование 233, 276
частотное разделение 70
частотное уплотнение 277
частотный план 334
червь сетевой 838
чередование байтов 323
четырехуровневая частотная манипуляция 258
чип 305
чиповая скорость 305
числовой адрес 59
чистая IP-сеть 689

Ш

Шеннона формула 228, 241, 246
ширина спектра сигнала 233
широковещательная рассылка 491
широковещательное радио 287
широковещательное сообщение 491
широковещательные услуги 147
широковещательный адрес 59, 360, 491, 528
широковещательный шторм 409, 491
шифрование
  понятие 848
  с открытым ключем 851
  с помощью односторонней функции 855
шлюз 810
  безопасности 894
  внешний 587
  понятие 483
  транспортный 815
шлюзовой протокол
  внешний 588
  внутренний 588
  пограничный 588
шпионская программа 844
шум дискретизации 263

Э

экран 876
экранированная витая пара 231, 247, 249
экстранет 684
эластичный трафик 189
электрическая связь 239
электронная подпись 859, 870
электронное сообщение 795
элементарный канал 78, 79, 262
элемент сети 821
эмуляция выполнения программ 876
эстафетная передача 295
эхо-запрос 596
эхо-ответ 596
эхо-протокол 596

Я

язык разметки гипертекста 802
ячейка 678
ячеистая топология 57, 324, 338

Виктор Григорьевич Олифер, Наталья Алексеевна Олифер

Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 4-е изд.

Руководитель проекта	*А. Юрченко*
Ведущий редактор	*Ю. Сергиенко*
Литературный редактор	*А. Жданов*
Художественный редактор	*Л. Адуевская*
Корректор	*В. Листова*
Верстка	*Е. Егорова*

Подписано в печать 08.10.09. Формат 70x100/16. Усл. п. л. 76,11. Тираж 4500. Заказ 19113.
ООО «Лидер», 194044, Санкт-Петербург, Б. Сампсониевский пр., д. 29а.
Налоговая льгота — общероссийский классификатор продукции ОК 005-93, том 2; 95 3005 — литература учебная.
Отпечатано по технологии CtP в ОАО «Печатный двор» им. А. М. Горького.
197110, Санкт-Петербург, Чкаловский пр., д. 15.

Учебник для вузов

Профессиональные биографии Виктора и Натальи Олифер очень похожи. Оба они получили свое первое высшее образование в МВТУ им. Н. Э. Баумана (специальность «Электронные вычислительные машины»), а второе — в МГУ им. М. В. Ломоносова (специальность «Прикладная математика»).

После защиты диссертации каждый из них совмещал преподавание в вузах с научно-исследовательской работой. В 1995 году Наталья и Виктор стали читать лекции по сетевым технологиям в Центре информационных технологий при МГУ. Ими были разработаны несколько авторских курсов, которые и составили в дальнейшем основу для написания популярных учебников «Сетевые операционные системы» и «Компьютерные сети», а также книги «Computer Networks: Principles, Technologies and Protocols for Network Design».

В настоящее время Наталья Олифер работает независимым консультантом в области сетевых технологий, а Виктор Олифер участвует в проекте по развитию сети JANET, объединяющей университеты и исследовательские центры Великобритании.

Новое издание одного из лучших российских учебников по сетевым технологиям можно считать юбилейным. Прошло ровно 10 лет с момента первой публикации книги «Компьютерные сети. Принципы, технологии, протоколы». За это время книга приобрела широкую популярность в России, была издана на английском, испанском, португальском и китайском языках, и с каждым новым изданием она существенно обновлялась. Не стало исключением и это, четвертое, издание, в котором появилось много новых разделов, посвященных самым актуальным направлениям сетевых технологий.

Издание предназначено для студентов, аспирантов и технических специалистов, которые хотели бы получить базовые знания о принципах построения компьютерных сетей, понять особенности традиционных и перспективных технологий локальных и глобальных сетей, изучить способы создания крупных составных сетей и управления такими сетями.

Рекомендовано Министерством образования и науки Российской Федерации в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлению «Информатика и вычислительная техника» и по специальностям «Вычислительные машины, комплексы, системы и сети», «Автоматизированные машины, комплексы, системы и сети», «Программное обеспечение вычислительной техники и автоматизированных систем».

ПИТЕР®

Заказ книг:

197198, Санкт-Петербург, а/я 127
тел.: (812) 703-73-74, postbook@piter.com

61093, Харьков-93, а/я 9130
тел.: (057) 758-41-45, 751-10-02, piter@kharkov.piter.com

ISBN: 978-5-49807-389-7

9 785498 073897

www.piter.com — вся информация о книгах и веб-магазин