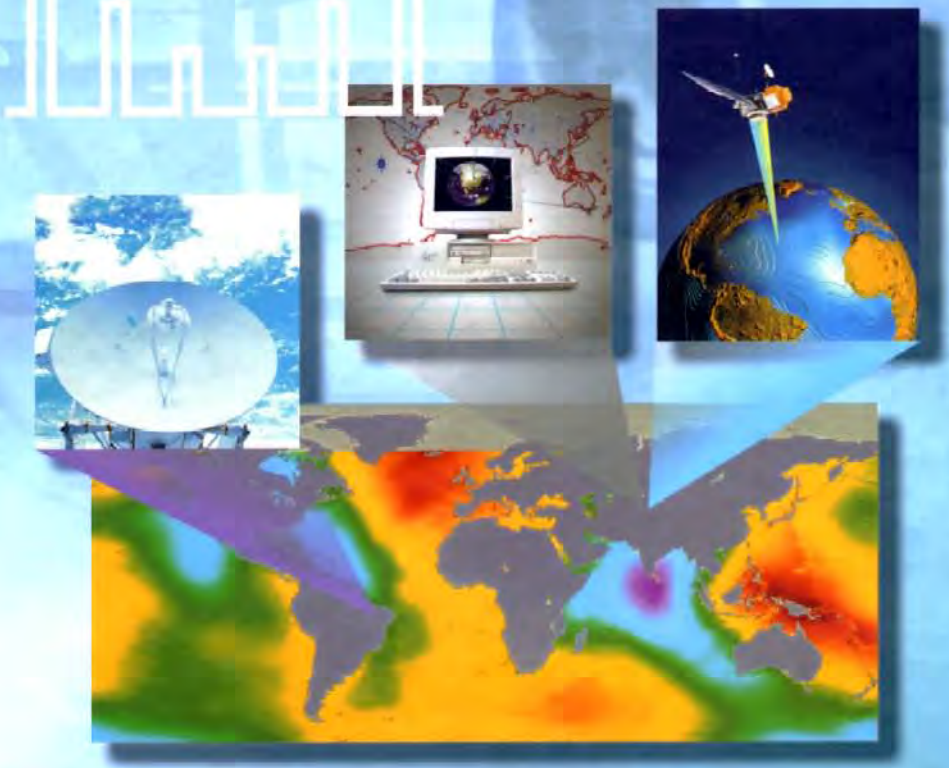
Информатика в техническом университете

В.А. Галкин, Ю.А. Григорьев

Телекоммуникации и сети



Издательство МГТУ имени Н.Э. Баумана

Информатика в техническом университете

Серия основана в 2000 году

РЕДАКЦИОННАЯ КОЛЛЕГИЯ:

В.А. Галкин, Ю.А. Григорьев

Телекоммуникации и сети

Допущено Министерством образования
Российской Федерации
в качестве учебного пособия
для студентов высших учебных заведений,
обучающихся по специальности
«Автоматизированные системы
обработки информации и управления»
направления подготовки
дипломированных специалистов
«Информатика и вычислительная техника»

Москва
Издательство МГТУ имени Н.Э. Баумана
2003

УДК 681.326(075)
ББК 32.973.202
Г161

Федеральная целевая программа «Культура России»
(подпрограмма «Поддержка полиграфии и книгоиздания России»)

Рецензенты:
кафедра «Автоматизированные системы управления» Московского автомобильно-дорожного института (д-р техн. наук, профессор А.Б. Николаев);
д-р техн. наук, профессор В.В. Соломенцев (зав. кафедрой «Вычислительные машины, комплексы, системы и сети» Московского государственного технического университета гражданской авиации)

Галкин В.А., Григорьев Ю.А.

Г161 Телекоммуникации и сети: Учеб. пособие для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2003. – 608 с.: ил. – (Сер. Информатика в техническом университете.)

ISBN 5–7038–1961–X

Приведены основы построения систем передачи данных и их характеристики, современные методы и технологии телекоммуникационных систем. Большое внимание в книге уделено методам построения локальных вычислительных сетей и сетевой операционной системе NetWare. Рассмотрены современные сетевые технологии (Frame relay, ISDN, ATM), мобильные и спутниковые сети, а также назначение и сценарии работы основных сетевых протоколов, вопросы объединения сетей и построения корпоративной электронной почты.

Содержание учебного пособия соответствует курсу лекций, который авторы читают в МГТУ им. Н.Э. Баумана.

Для студентов старших курсов технических вузов, специализирующихся в области автоматизированных систем обработки информации и компьютерных информационных технологий. Будет полезно специалистам в области средств телекоммуникаций и вычислительных сетей.

УДК 681.326(075)
ББК 32.973.202

ISBN 5–7038–1961–X

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ

В предлагаемом пособии описаны основы построения систем передачи данных и их характеристики, современные методы и технологии телекоммуникационных систем. Значительное внимание уделено методам построения локальных вычислительных сетей и сетевой операционной системе NetWare фирмы Novell. Подробно рассмотрены современные сетевые технологии (Frame Relay, ISDN, ATM), мобильные и спутниковые сети, а также назначение и сценарии работы основных сетевых протоколов, вопросы объединения сетей и построения корпоративной электронной почты.

Структура книги построена таким образом, чтобы обеспечить читателю как возможность получения справочной информации по интересующему вопросу, так и возможность изучения отдельных методов и технологий.

В главе 1 рассмотрены общие понятия распределенной обработки данных и концепции открытых систем.

Глава 2 посвящена основам телекоммуникации. В ней детально изложены современные методы и технологии передачи данных, принципы построения систем передачи дискретной информации, методы защиты от ошибок и сжатия данных. Здесь читатель найдет исчерпывающие сведения по технологиям передачи данных по телефонным каналам связи и техническим средствам телекоммуникаций.

В главе 3 рассмотрены принципы построения локальных сетей. Здесь приведена классификация локальных сетей и методов доступа, детально изложены технологии традиционных и высокоскоростных локальных сетей, а также методы построения виртуальных сетей. Приведено описание необходимого оборудования локальных сетей, его функций и основных характеристик.

Глава 4 посвящена описанию основных технологий глобальных сетей, таких как X.25, Frame Relay, ISDN, ATM, мобильных и спутниковых сетей. Здесь детально рассмотрены архитектура, протоколы и услуги сетей каждого типа.

В главе 5 изложены сетевые протоколы. В качестве основного стека протоколов для рассмотрения выбран стек протоколов TCP/IP как наиболее распространенный. В качестве протоколов IV уровня стека TCP/IP приведено описание протоколов HDLC, LAPD, SLIP и PPP. Уровень межсетевого взаимодействия представлен протоколами IPv4 и IPv6, ICMP и протоколами маршрутизации RIP и OSPF.

В главе 6 рассмотрены вопросы, посвященные принципам организации, адресации и маршрутизации в системах электронной почты. Здесь приведены различные классы систем, построенные на основе стандарта X.400, протокола SMTP, на основе частных стандартов cc:Mail и MS Mail, а также гибридные почтовые системы MS Exchange Server. Введены ключевые понятия системы электронных почтовых каталогов X.500 и связи каталогов почтовых систем различных классов.

Главы 7 – 9 посвящены сетевой операционной системе NetWare фирмы Novell.

Глава 10 освещает вопросы выбора архитектуры вычислительной сети. Здесь рассмотрены различные решения для локальных, городских и глобальных сетей, приведен сравнительный анализ различных сетевых технологий.

Учебное пособие предназначено для студентов старших курсов технических вузов, специализирующихся в области автоматизированных систем обработки информации и компьютерных информационных технологий, специалистам в области средств телекоммуникаций и вычислительных сетей.

Авторы глубоко признательны рецензентам: заведующему кафедрой «Вычислительные машины, комплексы, системы и сети» Московского государственного технического университета гражданской авиации докт. техн. наук, профессору В.В. Соломенцеву и коллективу кафедры «Автоматизированные системы обработки» МАДИ во главе с докт. техн. наук, профессором А.Б. Николаевым за полезные замечания и советы.

СПИСОК ОСНОВНЫХ СОКРАЩЕНИЙ

АДИКМ – адаптивная ДИКМ
АИС – автоматизированная информационная система
АПД – аппаратура передачи данных
АСИ – аппаратный связной интерфейс
АТС – автоматическая телефонная станция
АЦП – аналого-цифровой преобразователь
БД – база данных
ВОЛС – волоконно-оптическая линия связи
ВОС – взаимодействие открытых систем
ДВ – длинные волны
ДИКМ – дифференциальная ИКМ
ДК – дискретный канал
ДКП – дискретное косинусное преобразование, используется при сжатии неподвижного изображения по стандарту JPEG
ДН – диаграмма направленности
ИКМ – импульсно-кодовая модуляция
КВ – короткие волны
КТСОП – коммутируемая телефонная сеть общего пользования
МДКН – множественный доступ с контролем носителя
МДКН/ОС – множественный доступ с контролем носителя и обнаружением столкновений
МККТТ – Международный консультативный комитет по телеграфии и телефонии (современное название ITU-T)
МОС – Международная Организация Стандартов
ОГСС – общегородская справочная система
ООД – оконечное оборудование обработки данных

ОП	– оперативная памятьОП – оперативная память
ОС	– операционная система
ПЗУ	– постоянное запоминающее устройство
ПК	– персональный компьютер
ПО	– программное обеспечение
РСОД	– распределенная система обработки данных
СА	– сетевой адаптер
СБДСР	– системная база данных сетевых ресурсов
СВ	– средние волны
СВЧ	– диапазон сверхвысоких частот
СИ	– связные интерфейсы
СКС	– структурированная кабельная система
СПД	– сеть передачи данных
ССПС	– сеть сотовой подвижной связи
СУБД	– система управления базой данных
ТЧ	– канал тональной частоты
УКВ	– ультракороткие волны
УДС	– управление доступом к передающей среде
УЛЗ	– управление логическим звеном
ФМ	– фазовая модуляция
ФС	– подуровень физической сигнализации
ЦАП	– цифро-аналоговый преобразователь
ЦКП	– центр коммутации пакетов
ЦП	– центральный процессор
ЦУКС	– центр управления космическим сегментом
ЦУНС	– центр управления наземным сегментом
ЦУПУ	– центр управления поставщиков услуг
ЧМ	– частотная модуляция
ЭП	– электронная почта

ВВЕДЕНИЕ

Телекоммуникация и сетевые технологии являются в настоящее время той движущей силой, которая обеспечивает развитие мировой цивилизации. Практически нет области производственных и общественных отношений, которая не использовала бы возможности современных информационных технологий на базе телекоммуникаций.

Приступая к написанию данного учебного пособия, авторы исходили из того, что в данной динамически развивающейся области знаний уже существует много различных электронных публикаций на сайтах сети Интернет и печатных изданий, в том числе учебно-методической литературы. Поэтому в книгу включен только материал, который отработан на протяжении последних нескольких лет в учебном процессе кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана при чтении курсов «Вычислительные комплексы и сети», «Сети ЭВМ и телекоммуникации», «Сетевое программное обеспечение».

Информация, которой оперируют компьютеры, называется данными, а территориально распределенные и соединенные линиями связи компьютеры, занимающиеся обработкой данных, в общем случае представляют собой распределенную систему обработки данных или сеть ЭВМ. Линия связи – это либо телефонная линия, либо другая среда передачи данных: витая пара, волоконно-оптическая линия связи, коаксиальный кабель, радиолиния и т.п. Совокупность различных линий связи и каналообразующей аппаратуры представляется как телекоммуникационная среда, обеспечивающая удаленное взаимодействие компьютеров.

Основываясь на понятии архитектуры распределенных систем обработки данных, авторами в пособии сделана попытка изложить описание базовой модели взаимодействия открытых систем с позиций объектно-ориентированного подхода. Материал представлен в соответствии с уровнями эталонной модели, начиная с форм представления сигналов физических каналов связи.

Вопросы достоверности передаваемой информации в системах телекоммуникаций играют важную роль, поэтому им уделено значительное внимание. Подробно рассмотрены и проиллюстрированы примерами математические аспекты логического кодирования.

В качестве технических средств телекоммуникаций в учебном пособии представлены модемы, которые продолжают оставаться широко распространенными средствами удаленного взаимодействия и доступа к сети Интернет. Кроме того, модемы, пожалуй, единственные технические средства, которые доступны в настоящее время большинству вузов в качестве оборудования, обеспечивающего лабораторный практикум по соответствующим учебным курсам.

Рассмотрение сетевых технологий умышлено построено с нарушением хронологического порядка развития сетей ЭВМ. Как известно, глобальные сети и связанные с ними технологии появились раньше, чем локальные. Однако бурный рост локальных сетей и совершенствование их технологий за последние несколько лет определили в значительной степени это решение. Кроме того, авторы, не претендуя на оригинальность, посчитали методически правильным рассмотреть сначала методы построения локальных сетей и их технологий, тем более, что современные технологии, такие, как ATM и FDDI, стирают грани между понятиями «глобальный» и «локальный».

Представление о предмете было бы неполным без рассмотрения сетевых операционных систем. В связи с этим в книге даны понятия, определено назначение и рассмотрены основные характеристики и возможности сетевой операционной системы NetWare фирмы Novell.

Авторы надеются, что читатель приобретет некоторые практические навыки построения сетей на базе различных технологий, прочитав заключительную главу настоящего учебного пособия.

1. РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ ОБРАБОТКИ ДАННЫХ

Рассмотрены основные понятия и положения распределенной обработки данных. Описана архитектура распределенных систем как совокупность логической, физической и программной структур. Взаимодействие открытых систем рассмотрено с точки зрения объектно-ориентированного подхода. Достаточно подробно описаны принципы построения эталонной модели взаимодействия открытых систем OSI и функциональное назначение уровней этой модели. Кроме модели OSI приведены сведения об альтернативных профилях стандартов открытых систем.

1.1. Архитектура распределенных систем

Основные понятия распределенной обработки данных

Распределенная система обработки данных (РСОД) – любая система, позволяющая организовать взаимодействие независимых, но связанных между собой ЭВМ. Эти системы предназначены для автоматизации таких объектов, которые характеризуются территориальной распределенностью пунктов возникновения и потребления информации. Концептуально распределенная обработка подразумевает тот или иной вид организации сети связи и децентрализацию трех категорий ресурсов:

- аппаратных вычислительных средств и собственно вычислительной мощности;
- баз данных;
- управление системой.

В распределенных системах обработки данных в той или иной степени осуществляется реализация следующих основных функций:

- доступ к ресурсам (вычислительным мощностям, программам, данным и т. п.) с терминалов и из пользовательских программ в режиме «файл–сервер»;

• выполнение заданий и интерактивное общение пользователей с запущенными по их требованию программами в режиме «клиент–сервер»;
• сбор статистики о функционировании системы;
• обеспечение надежности и живучести системы в целом.

В настоящее время применяют различные подходы к классификации распределенных систем обработки данных по разным критериям.

По степени однородности различают:
• полностью неоднородные РСОД;
• частично неоднородные РСОД;
• однородные РСОД.

Полностью неоднородные РСОД характеризуются тем, что в них объединены ЭВМ, построенные на основе различных архитектур и функционирующие под управлением разных операционных систем (ОС). Как правило, РСОД этого типа в качестве коммуникационной службы используют глобальные сети, базирующиеся на протоколах X.25, Frame relay, ATM, Internet-технология.

Частично неоднородные РСОД строят на базе однотипных ЭВМ, работающих под управлением различных ОС, либо они включают в себя компьютеры различных типов, работающие под управлением одной ОС. Например, IBM PC компьютеры управляются различными ОС: MS DOS, OS/2, Windows 95, Windows NT.

Однородные распределенные системы строятся на однотипных вычислительных средствах, оснащенных одинаковыми операционными системами.

По архитектурным особенностям выделяют:
• РСОД на основе систем телеобработки;
• РСОД на основе сетевой технологии.

Под сетевой технологией понимается такая форма взаимодействия ЭВМ, при которой любой из процессов одной из машин по своей инициативе может установить логическую связь с любым процессом в любой другой ЭВМ.

В отличие от таких систем РСОД на основе систем телеобработки не обеспечивают полного, симметричного и независимого взаимодействия процессов.

По степени распределенности с позиций пользователя РСОД делятся на 2 группы: региональные и локальные.

К региональным РСОД будем относить распределенные конфигурации, характеризующиеся следующими основными параметрами:

неограниченной географической распределенностью;

наличием тех или иных механизмов маршрутизации;

каждые два узла связаны собственным каналом, и отсутствует проблема его разделения;

широким диапазоном скоростей передачи – 10^3... 10^8 бит/с;

произвольной топологией.

В них можно выделить несколько способов организации взаимодействия между ЭВМ:

коммутация каналов;

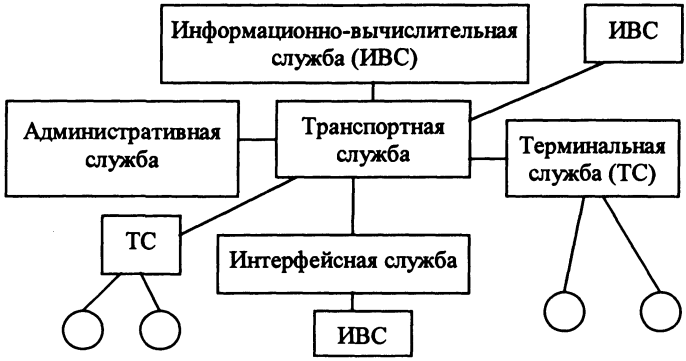


Рис. 1.1. Логическая структура РСОД

коммутация сообщений;

коммутация пакетов;

коммутация фреймов – Frame relay;

коммутация ячеек – ATM-технология.

Основу локальных РСОД составляют локальные сети со следующими характеристиками:

небольшая географическая распределенность;

использование единой коммуникационной среды и, следовательно, физическая полносвязность всех узлов сети, приводящая к замене маршрутизации адресацией;

высокие и очень высокие скорости обмена – 10^7... 10^9 бит/с;

применение специальных методов и алгоритмов доступа к единой среде для обеспечения высокой скорости передачи при одновременном использовании среды всеми узлами коммуникационной службы;

ограниченность возможных топологий.

Под архитектурой РСОД будем понимать взаимосвязь ее логической, физической и программной структур.

Логическая структура РСОД отражает состав сетевых служб и связи между ними (рис. 1.1). В данной структуре информационно-вычислительная служба предназначена для решения задач пользователей сети. Терминальная служба обеспечивает взаимодействие терминалов с сетью. Сюда входит преобразование форматов и кодов, управление разнотипными терминалами, обработка процедур обмена информацией между терминалами и сетью и т. д. Транспортная служба предназначена для решения всех задач, связанных с передачей сообщений в сети. Она управляет маршрутами, потоками и данными, декомпозицией сообщений на пакеты и рядом других функций. Интерфейсная служба решает задачи обеспечения взаимодействий разнотипных ЭВМ, функционирующих под управлением различных ОС, имеющих разную архитектуру, длину слова, форматы представления данных и др. Кроме того, служба управления

интерфейсами осуществляет взаимодействие ЭВМ, входящих в состав различных сетей. Административная служба управляет сетью, реализует процедуры реконфигурации и восстановления, собирает статистику о функционировании сети, осуществляет ее тестирование. Разумеется, приведенный полный состав элементов логической структуры не является обязательным для всех реальных систем. Так, в однородных сетях отпадает необходимость в интерфейсной службе, в простейших сетях может отсутствовать административная служба и т. д. Информационно-вычислительная и терминальная службы образуют *абонентскую службу*, а интерфейсная и транспортная – *коммуникационную*. Из этого следует, что административная служба не осуществляет непосредственно какие-либо функции, связанные с сетевым обслуживанием пользователей, и может рассматриваться как механизм обслуживания самой сети. Распределение элементов логической структуры по различным ЭВМ задает *физическую структуру* РСОД (рис. 1.2).

Элементами такой структуры являются ЭВМ, связанные между собой и с терминалами. В зависимости от реализации в ЭВМ той или иной сетевой службы в физической структуре можно выделить:

главные ЭВМ;
коммуникационные ЭВМ;
интерфейсные ЭВМ;
терминальные ЭВМ;
административные ЭВМ.

В одной ЭВМ могут реализовываться несколько служб.

Программная структура РСОД отражает состав компонентов сетевого программного обеспечения (ПО) и связи между ними. Очевидно, что состав сетевого ПО определяется логической структурой, т. е. функциями, выполняемыми ее службами. В то же время связи между компонентами ПО во многом зависят от физической структуры.

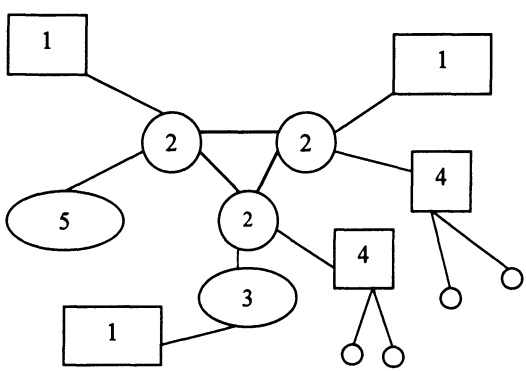


Рис. 1.2. Физическая структура РСОД

Сетевое ПО имеет многоуровневую иерархическую организацию. Что обусловлено двумя факторами:

- необходимостью минимизации затрат на модификацию сетевого ПО при изменении состава используемого оборудования;
- любые осуществляемые в сети изменения не должны отражаться на пользовательских программах, использующих сетевые возможности.

Для иерархической организации необходимы правила взаимодействия программ, выполняемых в одной ЭВМ и находящихся на различных уровнях, и программ, находящихся на одном уровне, но расположенных в различных ЭВМ, т. е. четкое описание *интерфейсов* и *протоколов*.

Стремление создать единую, универсальную и открытую к изменениям логической и физической структур сетевую архитектуру обусловило стандартизацию уровней иерархии ПО сетей ЭВМ. Международная организация по стандартизации (ISO – International Standard Organization) предложила концепцию архитектуры открытых систем, в которой определена эталонная модель, используемая как базовая при разработке международных стандартов.

1.2. Концепция открытых систем

В соответствии с эталонной моделью вычислительная сеть представляется как распределенная информационно-вычислительная среда, реализуемая большим числом разнообразных аппаратных и программных средств. Эта среда по вертикали делится на ряд логических уровней, каждый из которых выполняет одну из основных задач информационно-вычислительной среды. По горизонтали она делится на локальные части , называемые открытыми системами, каждая из которых удовлетворяет требованиям и стандартам архитектуры открытых систем ISO (рис. 1.3). Термин «взаимодействие открытых систем» ВОС (OSI – Open System Interconnection) относится к процедурам передачи данных между системами, которые «открыты» друг другу благодаря совместному использованию ими соответствующих стандартов.

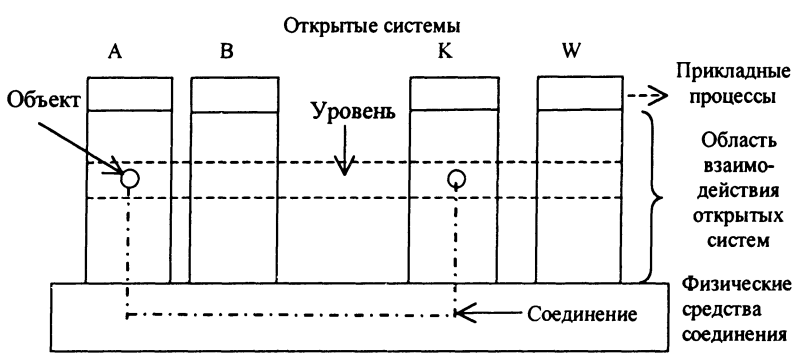


Рис. 1.3. Область взаимодействия открытых систем

Сложность функций области взаимодействия привела к тому, что они в соответствии с базовой эталонной моделью взаимодействия открытых систем поделены на семь расположенных друг над другом слоев, называемых *уровнями*. Их иерархия в зависимости от выполняемых функций делится на две части – прикладную и транспортную платформы.

Часть открытой системы, реализующая некоторую функцию и входящая в состав того или иного уровня, называется *объектом*. Набор правил взаимодействия объектов одного и того же N-го уровня называется N-протоколом. Связь между объектами соседних уровней определяется *интерфейсом* (например, связь между объектами N- и $(N-1)$-го уровней определяется $(N-1)$-м интерфейсом.

Общие свойства открытых систем обычно формируются следующим образом:

- расширяемость/масштабируемость – extensibility/scalability,
- мобильность (переносимость) – portalility,
- интероперабельность (способность к взаимодействию с другими системами) – interoperability,
- дружественность к пользователю, в том числе легкая управляемость – driveability.

Эти свойства, взятые по отдельности, были характерны и предыдущим поколениям информационных систем и средств вычислительной техники. Новый взгляд на открытые системы определяется тем, что эти свойства рассматриваются в совокупности, как взаимосвязанные, и реализуются в комплексе.

Взаимодействие открытых систем стало основной концепцией архитектуры информационных сетей.

Понятие «система» носит двойственный характер. С одной стороны, по общему определению, система – это совокупность взаимодействующих элементов (компонентов), аппаратных и/или программных. С другой стороны, система может выступать в качестве компонента другой, более сложной системы, которая, в свою очередь, может быть компонентом системы следующего уровня. В связи с этим нужно уточнить представление об архитектуре систем и средств как внешнем их описании (reference model) с точки зрения того, кто ими пользуется. Архитектура открытой системы, таким образом, оказывается иерархическим описанием ее внешнего облика и каждого компонента с точки зрения:

- пользователя (пользовательский интерфейс);
- проектировщика системы (среда проектирования);
- прикладного программиста (системы и инструментальные средства /среды программирования);
- системного программиста (архитектура ЭВМ);
- разработчика аппаратуры (интерфейсы оборудования).

Для примера рассмотрим архитектурное представление системы обработки данных, состоящей из четырех компонентов: пользовательского интерфейса (соответственно точкам зрения всех указанных выше групп), средств обработки данных, средств представления и хранения данных, средств коммуникаций. Для этого представления необходимо три уровня описаний: среды, которая представляется системой, операционной среды (системы), на которую опираются прикладные компоненты, и оборудования. Каждый из этих уровней для удобства разделен на два подуровня (табл. 1.1).

Уровень среды для конечного пользователя (user environment) характеризуется входными и выходными описаниями (генераторы форм и отчетов), языками проектирования информационной модели предметной области (языки 4GL), функциями утилит и библиотечных программ и прикладным уровнем среды коммуникаций, когда необходимы услуги дистанционного обмена информацией. На этом же уровне определена среда (инструментарий) прикладного программирования (application environment): языки и системы программирования, командные языки (оболочки ОС), языки запросов систем управления базами данных (СУБД), уровни сессий и представительный среды коммуникаций.

Таблица 1.1. **Уровни архитектуры системы обработки данных**

Уровень архитектуры системы обработки данных	Компоненты системы обработки данных			
	Интерфейсы	Средства обработки данных	Представление и хранение данных	Коммуникации в модели OSI
Среда для конечного пользователя и инструментарий прикладного программиста	Генераторы форм и отчетов	Утилиты и библиотеки	Языки программирования 4GL	Прикладной уровень
	Языки программные и командные (оболочки)	Прикладные программы	Языки запросов СУБД	Уровни сессий и представительный
Операционная система	Средства оконного интерфейса	Верхний уровень ОС (организация процесса обработки)	Средства доступа к среде хранения	Транспортный уровень
	Драйверы	Ядро операционной системы	Файловая система	Сетевой уровень
Оборудование	Системные интерфейсы (в том числе организация ввода-вывода)	Процессоры (система команд)	Организация памяти	Уровень передачи данных
	Периферийные устройства	Системная шина	Шины (интерфейс) массовой памяти	Физический уровень

На уровне ОС представлены компоненты операционной среды, реализующие функции организации процесса обработки, доступа к среде хранения данных, оконного интерфейса, а также транспортного уровня среды коммуникаций. Нижний подуровень ОС – это ее ядро, файловая система, драйверы управления оборудованием, сетевой уровень среды коммуникаций.

На уровне оборудования четко видны привычные разработчикам ЭВМ составляющие архитектуры аппаратных средств:

- система команд процессора (процессоров),
- организация памяти,
- организация ввода-вывода и т. д.,

а также физическая реализация в виде:

системных шин;
шин массовой памяти;
интерфейсов периферийных устройств;
уровня передачи данных;
физического уровня среды хранения.

Представленный взгляд на архитектуру открытой системы обработки данных относится к одномашинным реализациям, включенным в сеть передачи данных для обмена информацией. Понятно, что он может быть легко обобщен и на многопроцессорные системы с разделением функций, а также на РСОД. Поскольку здесь явно выделены компоненты, составляющие систему, можно рассматривать интерфейсы взаимодействия этих компонентов на каждом из указанных уровней и интерфейсы взаимодействия между уровнями. Описания и реализации этих интерфейсов могут быть предметом рассмотрения только в пределах данной системы. Тогда свойства ее открытости проявят только на внешнем уровне. Однако значение идеологии открытых систем состоит в том, что она открывает методологические пути к унификации интерфейсов в пределах родственных по функциям групп компонентов для всего класса систем данного назначения или всего множества открытых систем. Область распространения этих стандартов является предметом согласования интересов разных групп участников процесса информатизации – пользователей, проектировщиков систем, поставщиков программных продуктов и поставщиков оборудования.

Преимущества идеологии открытых систем. Конечно, подход открытых систем пользуется успехом только потому, что обеспечивает преимущества для различных специалистов, связанных с областью компьютеров.

Для пользователя открытые системы обеспечивают следующее:

- новые возможности сохранения сделанных вложений благодаря свойствам эволюции, постепенного развития функций систем, замены отдельных компонентов без перестройки всей системы;
- независимость от поставщиков аппаратных или программных средств, возможность выбора продуктов из предложенных на рынке при условии соблюдения поставщиком соответствующих стандартов открытых систем;

• дружественность среды, в которой работает пользователь, мобильность персонала в процессе эволюции системы;

• возможность использования информационных ресурсов, имеющихся в других системах (организациях).

Проектировщик информационных систем получает:

• возможность использования разных аппаратных платформ;

• возможность совместного использования прикладных программ, реализованных в разных ОС;

• развитые средства инструментальных сред, поддерживающих проектирование;

• возможность использования готовых программных продуктов и информационных ресурсов.

Разработчики общесистемных программных средств имеют:

• новые возможности разделения труда, благодаря повторному использованию программ (reusability);

• развитые инструментальные среды и системы программирования;

• возможности модульной организации программных комплексов благодаря стандартизации программных интерфейсов.

Последнее свойство открытых систем позволяет пересмотреть традиционно сложившееся дублирование функций в разных программных продуктах, из-за чего системы, интегрирующие эти продукты, непомерно разрастаются по объему, теряют эффективность. Известно, что в одной и той же области обработки данных и текстов многие продукты, предлагаемые на рынке (текстовые редакторы, настольные издательства, электронные таблицы, системы управления базами данных) по ряду функций дублируют друг друга, а иногда и подменяют функции ОС. Кроме того, замечено, что в каждой новой версии этих продуктов их размеры увеличиваются на 15 %.

В распределенных системах, содержащих несколько рабочих мест на персональных компьютерах и серверов в локальной сети, избыточность программных кодов из-за дублирования возрастает многократно. Идеология и стандарты открытых систем позволяют по-новому взглянуть на распределение функций между программными компонентами систем и значительно повысить тем самым эффективность. Частично этот подход обеспечивает компенсацию затрат ресурсов, которые приходится платить за преимущества открытых систем относительно закрытых, ресурсы которых в точности соответствуют задаче, решаемой системой.

Открытые системы и объектно-ориентированный подход

В связи с применением открытых систем весьма перспективным направлением представляется объектно-ориентированный подход проектирования и программирования.

Объектно-ориентированное программирование – это относительно новый подход к разработке программных систем, строящийся по следующим основным принципам:

данные и процедуры объединяют в программные объекты;

для связи объектов используют механизм посылки сообщения;

объекты с похожими свойствами объединяют в классы;

объекты наследуют свойства других объектов через иерархию классов.

Объектно-ориентированные системы обладают следующими основными свойствами:

• *инкапсуляция* (скрытие реализации) – данные и процедуры объекта скрываются от внешнего пользователя, и связь с объектом ограничивается набором сообщений, которые «понимает» объект;

• *полиморфизм* (многозначность сообщений) – одинаковые сообщения поразному понимаются разными объектами, в зависимости от их класса;

• *динамическое (позднее) связывание* – значение имени (область памяти для данных или текст программы для процедур) становится известным только во время выполнения программы;

• *абстрактные типы данных* – объединение данных и операций для описания новых типов, позволяющие использовать новые типы наравне с уже существующими.

• *Наследование* – позволяет при создании новых объектов использовать свойства уже существующих объектов, описывая заново только те свойства, которые отличаются.

Объектно-ориентированный подход реализации системы хорошо согласуется с основными свойствами открытых систем (табл. 1.2).

Мобильность. Инкапсуляция позволяет хорошо скрыть машинно-зависимые части системы, которые должны быть реализованы заново при переходе на другую платформу. При этом гарантируется, что остальная часть системы не потребует изменений.

Таблица 1.2. **Свойства открытых систем и объектно-ориентированных систем программирования**

Свойства открытых систем	Дружественность (пользователь)	Мобильность (платформы)	Расширяемость (новые функции и области применения)	Интероперабельность (другие системы, пользователь)
Свойства объектно-ориентированных систем программирования	Объектное представление предметной области, наиболее удобное человеку. Сочетание всех других свойств при конструировании пользовательского интерфейса	Инкапсуляция (скрытие реализации)	Наследование, абстрактные типы данных	Полиформизм, динамическое связывание

При реализации новых машинно-зависимых частей многое может быть взято из уже существующей системы благодаря механизму наследования.

Расширяемость. Наследование позволяет сэкономить значительные средства при расширении системы, поскольку многое не нужно создавать заново, а некоторые новые компоненты можно получить, лишь слегка изменив старые. Использование отлаженных компонентов увеличивает надежность.

Возможность конструирования абстрактных типов данных для создания новых средств обеспечивается самим понятием класса, объединяющего похожие объекты с одинаковым набором операций.

Интероперабельность. Способность системы взаимодействовать с другими системами базируется на принципе посылки сообщения и соответствующих понятиях полиморфизма и динамического связывания. В сообщении объекту (возможно удаленному) передается имя действия, которое должно быть им выполнено, и некоторые дополнительные аргументы сообщения. Как это действие выполнять знает и решает только сам объект-получатель сообщения. От него только требуется выдать в ответ результат. Совершенно очевидно, что различные объекты будут по-разному реагировать на одинаковые сообщения (полиморфизм). Кроме того, очень удобно выбирать способ реализации в последний момент (при ответе на сообщение) в зависимости от текущего состояния системы (динамическое связывание).

Для того, чтобы разные системы могли обмениваться сообщениями, необходима либо единая трактовка всех типов данных, в том числе абстрактных, либо индивидуальная процедура преобразования сообщения для каждой пары неодинаковых взаимодействующих систем. Простота понятия абстрактных типов данных в объектно-ориентированных системах существенно облегчает разработку такой процедуры.

Дружественность. Удобство взаимодействия человека с системой требует от последней наличия всех трех вышеуказанных качеств. Мобильность необходима ввиду быстрой смены старых и появления новых устройств, в частности, средств мультимедиа. Расширяемость необходима для разработки программной поддержки новых парадигм общения человека с машиной. Интероперабельность рассматривает человека как другую систему, с которой открытая система должна уметь взаимодействовать.

1.3. Эталонная модель взаимодействия открытых систем

Базовая эталонная модель OSI является концептуальной основой, определяющей характеристики и средства открытых систем. Она определяет взаимодействие открытых систем, обеспечивающее работу в одной сети систем, выпускаемых различными производителями, и координирует:

- взаимодействие прикладных процессов;
- формы представления данных;
- единообразное хранение данных;
- управление сетевыми ресурсами;
- безопасность данных и защиту информации;
- диагностику программ и технических средств.

Модель разработана международной организацией стандартов (МОС) – ISO и широко используется во всем мире как основа концепций информационных сетей и их ассоциации. На базе этой модели задаются правила и процедуры передачи данных между открытыми системами. Рассматриваемая модель также описывает структуру открытой системы и комплексы стандартов, которым она должна удовлетворять. Основными элементами модели являются уровни, объекты, соединения, физические средства соединения.

В модели OSI средства взаимодействия делятся на семь уровней: прикладной, представительный, сеансовый, транспортный, сетевой, канальный и физический. Каждый уровень имеет дело с одним определенным аспектом взаимодействия сетевых устройств (рис. 1.4).

Модель OSI описывает только системные средства взаимодействия, реализуемые ОС, системными утилитами, системными аппаратными средствами. Модель не включает в себя средства взаимодействия приложений конечных пользователей. Свои собственные протоколы взаимодействия приложения реализуют, обращаясь к системным средствам. Поэтому нужно различать уровень взаимодействия приложений и прикладной уровень.

Необходимо также иметь в виду, что приложение может взять на себя функции некоторых верхних уровней модели OSI. 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 напрямую к системным средствам, ответственным за транспортировку сообщений по сети, которые располагаются на нижних уровнях модели.

Пусть приложение обращается с запросом к прикладному уровню, например к файловой службе. На основании этого запроса ПО прикладного уровня формирует сообщение стандартного формата. Обычное сообщение состоит из заголовка и поля данных. Заголовок содержит служебную информацию, которую необходимо передать через сеть прикладному уровню машины-адресата, чтобы сообщить ему, какую работу надо выполнить. В нашем примере заголовок, очевидно, должен содержать информацию о месте нахождения файла и о типе операции, которую необходимо над ним выполнить. Поле данных сообщения может быть пустым или содержать какие-либо данные, например те, которые необходимо записать в удаленный файл. Для того, чтобы доставить эту информацию по назначению, предстоит решить еще много задач, ответственность за которые несут нижележащие уровни модели OSI.

После формирования сообщения прикладной уровень направляет его вниз по стеку представительному уровню. Протокол представительного уровня на основании информации, полученной из заголовка прикладного уровня, выполняет требуемые действия и добавляет к сообщению собственную служебную информацию – заголовок представительного уровня, в котором содержатся указания для протокола представительного уровня машины-адресата. Полученное в результате сообщение передается вниз сеансовому уровню, который, в свою очередь, добавляет свой заголовок, и т. д. Наконец, сообщение дости-

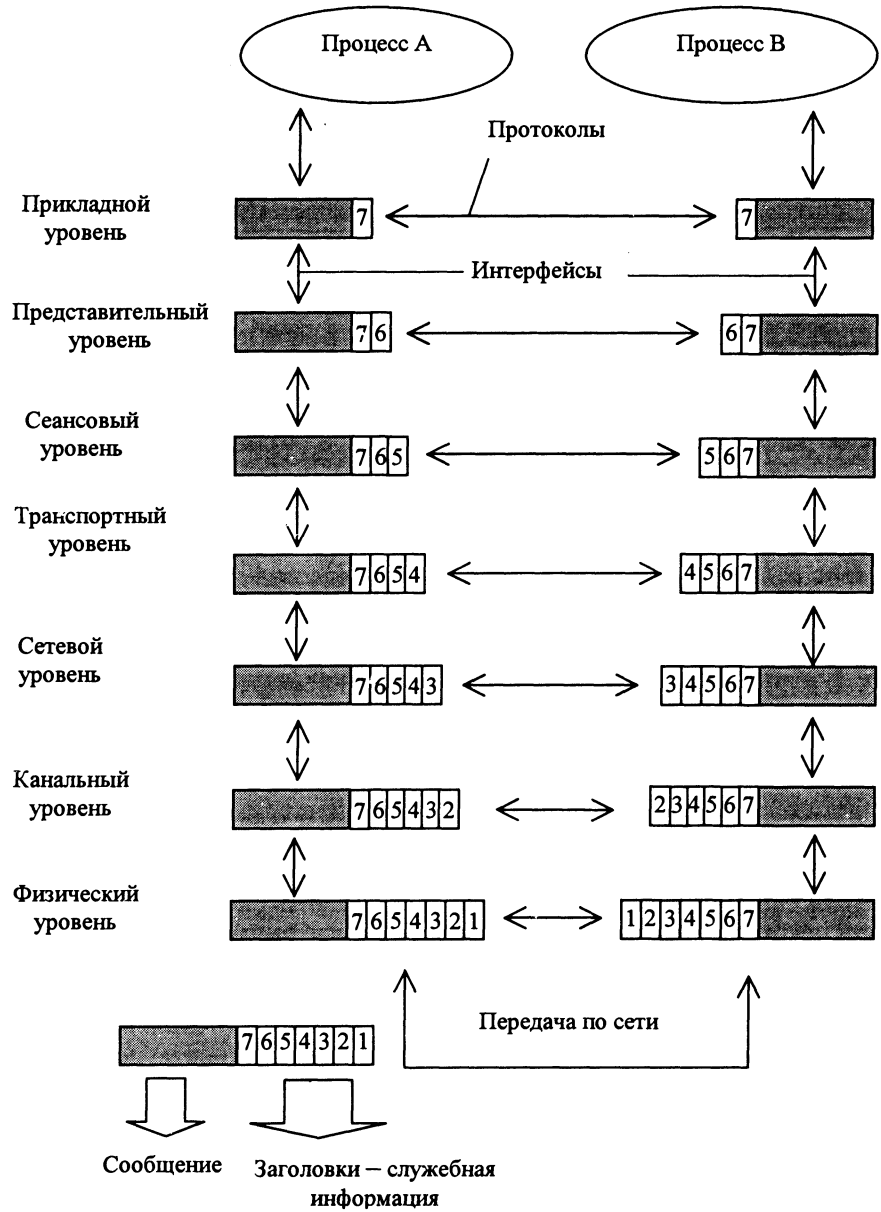


Рис. 1.4. Модель взаимодействия открытых систем OSI/ISO

гает самого нижнего, физического уровня, который собственно и передает его по линиям связи машине-адресату. К этому моменту сообщение содержит заголовки и концевики всех уровней (рис. 1.5).

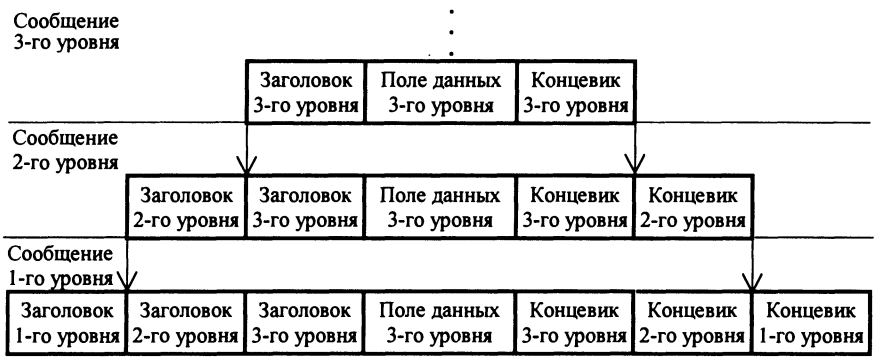


Рис. 1.5. Вложенность сообщений различных уровней

При поступлении сообщения на машину-адресат ее физический уровень принимает его. Далее оно последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует и обрабатывает заголовок своего уровня, выполняя соответствующие данному уровню функции, а затем удаляет этот заголовок и посылает сообщение вышележащему уровню.

Для обозначения единиц данных, с которыми имеют дело протоколы разных уровней, в OSI используется общее название *протокольный блок данных* (PDU – Protocol Data Unit). Для обозначения блоков данных определенных уровней часто используют специальные названия: кадр (frame), пакет (packet), дейтаграмма (datagram), сегмент (segment).

В модели OSI различаются два основных типа протоколов:

- с установлением соединения (connection-oriented);
- без предварительного установления соединения (connectionless).

В протоколах *с установлением соединения* перед обменом данными отправитель и получатель должны сначала установить соединение и, возможно, выбрать некоторые параметры протокола, которые они будут использовать при обмене данными. После завершения диалога они должны разорвать это соединение.

Протоколы *без предварительного установления соединения* называют также *дейтаграммными* протоколами. Отправитель просто передает сообщение, когда оно готово. Опускание письма в почтовый ящик является примером связи без предварительного установления соединения. При взаимодействии ЭВМ в РСОД используют протоколы обоих типов.

Уровни модели OSI

Физический уровень. Физический уровень (Physical layer) имеет дело с передачей битов по физическим каналам связи, таким, например, как коаксиальный кабель, витая пара, оптоволоконный кабель или радиосреда. К этому уровню имеют отношение характеристики физических сред передачи данных, такие, как полоса пропускания, помехозащищенность, затухание и др. На этом

же уровне определяют характеристики электрических сигналов, передающих дискретную информацию, например, крутизну фронтов импульсов, уровни напряжения или тока передаваемого сигнала, скорость передачи сигналов и типы кодирования. Здесь же стандартизуют типы разъемов и назначение каждого контакта.

Функции физического уровня реализуют все устройства, подключенные к сети. Со стороны, например, персональной ЭВМ (компьютера) функции физического уровня выполняет сетевой адаптер или последовательный коммуникационный порт.

Канальный уровень. Одной из задач канального уровня (Data Link layer) является проверка доступности среды передачи, так как физическая среда может быть занята одной из нескольких пар попеременно взаимодействующих компьютеров. Другой – реализация механизмов обнаружения и коррекции ошибок. Для этого на канальном уровне биты группируются в наборы, называемые *кадрами.* Канальный уровень обеспечивает корректность передачи каждого кадра, для выделения обрамляя его специальной последовательностью битов, а также вычисляет контрольную последовательность, добавляя ее к кадру.

При получении кадра адресат снова вычисляет контрольную последовательность. Если принятая с кадром и вычисленная контрольные последовательности совпадают, кадр считается правильным и принимается. Если же они не совпадают, то фиксируется ошибка. Канальный уровень может не только обнаруживать ошибки, но и исправлять их за счет повторной передачи поврежденных кадров. Необходимо отметить, что функция исправления ошибок не является обязательной для канального уровня, поэтому в некоторых протоколах этого уровня она отсутствует, например, в Ethernet и Frame relay.

В компьютерах локальных сетей функции канального уровня реализуются совместными усилиями сетевых адаптеров и их драйверов. Канальный уровень обеспечивает доставку кадра между любыми двумя узлами локальной сети той топологии, для которой он был разработан. К таким типовым топологиям, поддерживаемым протоколами канального уровня локальных сетей, относятся *общая шина*, *кольцо* и *звезда*, а также структуры, полученные с помощью мостов и коммутаторов. Примерами протоколов канального уровня являются протоколы Ethernet, Token Ring, FDDI, 100VG-AnyLAN.

В глобальных сетях, которые редко обладают регулярной топологией, канальный уровень часто обеспечивает обмен сообщениями только между двумя соседними компьютерами, соединенными индивидуальной линией связи. Примерами протоколов «точка-точка» (так часто называют такие протоколы) могут служить широко распространенные протоколы PPP и LAP-B.

Сетевой уровень. Сетевой уровень (Network layer) служит для образования единой транспортной системы, объединяющей несколько сетей, причем эти сети могут использовать различные принципы передачи сообщений между конечными узлами и обладать произвольной структурой связей.

На сетевом уровне термин «сеть» наделяют специфическим значением. В данном случае под сетью понимают совокупность компьютеров, соединенных между собой в соответствии с одной из стандартных типовых топологий и использующих для передачи данных один из протоколов канального уровня, определенный для этой топологии.

Внутри сети доставка данных обеспечивается соответствующим канальным уровнем, а вот доставкой данных между сетями занимается сетевой уровень, который и поддерживает возможность правильного выбора маршрута передачи сообщения даже в том случае, когда структура связей между составляющими сетями имеет характер, отличный от принятого в протоколах канального уровня.

Сети соединены между собой специальными устройствами, называемыми маршрутизаторами. *Маршрутизатор* – устройство, которое собирает информацию о топологии межсетевых соединений и на ее основании пересылает пакеты сетевого уровня в сеть назначения. Проблема выбора наилучшего пути называется *маршрутизацией*, и ее решение является одной из главных задач сетевого уровня. Данная проблема осложняется тем, что самый короткий путь не всегда самый лучший. Часто критерием при выборе маршрута является время передачи данных по этому маршруту; оно зависит от пропускной способности каналов связи и интенсивности трафика, которая может изменяться с течением времени. Некоторые алгоритмы маршрутизации стремятся адаптироваться к изменению нагрузки, в то время как другие принимают решения на основе средних показателей за длительное время. Выбор маршрута может осуществляться и по другим критериям, например надежности передачи.

Сетевой уровень решает также задачи согласования разных технологий, упрощения адресации в крупных сетях. Он отвечает за адресацию сообщений и перевод логических адресов и имен в физические адреса. Одним словом, исходя из конкретных сетевых условий, приоритета услуги и других факторов здесь определяется маршрут от компьютера-отправителя к компьютеру-получателю.

На этом уровне решаются также такие задачи и проблемы, связанные с сетевым трафиком, как коммутация пакетов и перегрузки. Если транзитная сеть не может передавать большие блоки данных, посланные компьютером-отправителем, то на сетевом уровне эти блоки разбиваются на меньшие, а сетевой уровень компьютера-получателя собирает эти данные в исходное состояние.

Сообщения сетевого уровня принято называть *пакетами* (packets). При организации доставки пакетов на сетевом уровне используют понятие «номер сети». В этом случае адрес получателя состоит из старшей части – номера сети и младшей – номера узла в этой сети. Все узлы одной сети должны иметь одну и ту же старшую часть адреса, поэтому термину «сеть» на сетевом уровне можно дать более формальное определение: сеть – совокупность узлов, сетевой адрес которых содержит один и тот же номер сети.

На сетевом уровне работают два вида протоколов. Первый вид – *сетевые протоколы* (routed protocols) – реализуют продвижение пакетов через сеть, второй – протоколы обмена маршрутной информацией или просто *протоколы маршрутизации* (routing protocols). С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений.

Протоколы сетевого уровня реализуются программными модулями ОС, а также программными и аппаратными средствами маршрутизаторов.

На сетевом уровне работают протоколы, которые отвечают за перевод логического адреса узла, используемого на сетевом уровне, в физический адрес, используемый в локальной сети. Это так называемые *протоколы разрешения адресов* (ARP – Address Resolution Protocol).

Примерами протоколов сетевого уровня являются протокол межсетевого взаимодействия IP стека TCP/IP и протокол обмена пакетами IPX стека Novell.

Транспортный уровень. На пути от отправителя к получателю пакеты могут быть искажены или утеряны, однако, некоторые приложения предпочитают иметь дело с надежным соединением. Поэтому основной функцией транспортного уровня (Transport layer) является обеспечение гарантированной доставки пакетов без ошибок, в той же последовательности, без потерь и дублирования. На этом уровне сообщения переупаковываются: длинные разбиваются на несколько пакетов, короткие объединяются в один. Это увеличивает эффективность передачи пакетов по сети. На транспортном уровне компьютера-получателя сообщения распаковываются, восстанавливаются в первоначальном виде и обычно посылается сигнал подтверждения приема.

Транспортный уровень управляет потоком, проверяет ошибки и участвует в решении проблем, связанных с отправкой и получением пакетов. Модель OSI определяет пять классов сервиса, предоставляемых транспортным уровнем. Эти классы 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, а 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, например, 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 ниж-

них уровней изначально очень ненадежны, то целесообразно обратиться к наиболее развитому сервису транспортного уровня, который работает, используя максимум средств для обнаружения и устранения ошибок – с помощью предварительного установления логического соединения, контроля доставки сообщений по контрольным суммам , контроля упорядоченной доставки пакетов с использованием их циклической нумерации, использование механизма таймаута и т. п.

Как правило, все протоколы, начиная с транспортного уровня и выше, реализуются программными средствами конечных узлов сети – компонентами их сетевых ОС. 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. Остальные три верхних уровня решают задачи предоставления прикладных сервисов на основании имеющейся транспортной подсистемы.

Сеансовый уровень. Сеансовый уровень (Session layer) позволяет двум приложениям на разных компьютерах устанавливать, использовать и завершать соединение, называемое сеансом. На этом уровне выполняются такие функции, как распознавание имен и защита, необходимые для связи двух приложений в сети, обеспечивает управление диалогом между взаимодействующими процессами, т. е. регулируется, какая из сторон осуществляет передачу, когда, как долго и т. д.

Сеансовый уровень обеспечивает синхронизацию между пользовательскими заданиями посредством расстановки в потоке данных контрольных точек (checkpoints). Таким образом, в случае сетевой ошибки, потребуется заново передать только данные, следующие за последней контрольной точкой.

На практике немногие приложения используют сеансовый уровень, и он редко реализуется в виде отдельных протоколов, хотя функции этого уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе.

Представительный уровень. Представительный уровень (Presentation layer) определяет форму обмена данными между сетевыми компьютерами. Этот уровень является переводчиком. На компьютере-отправителе данные, поступившие от прикладного уровня, переводятся в общепонятный промежуточный формат. На компьютере-получателе на этом уровне происходит перевод из промежуточного формата в тот, который используется прикладным уровнем данного компьютера, т. е. представительный уровень имеет дело с формой представления 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.

Представительный уровень отвечает за преобразование протоколов, трансляцию данных, замену или преобразование применяемого набора символов (кодовой таблицы) и расширение графических команд. Представительный уровень, кроме того, управляет сжатием данных для уменьшения передаваемых битов.

На этом уровне может выполняться шифрование и дешифрование данных, благодаря которому секретность обмена данными обеспечивается сразу для всех прикладных служб. Примером такого протокола является протокол Secure Socket Layer (SSL), который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня стека TCP/IP.

На представительном уровне работает утилита, называемая редиректором (redirector). Ее назначение – переадресовать операции ввода/вывода к ресурсам сервера.

Прикладной уровень. Прикладной уровень (Application layer) – самый верхний уровень модели – представляет собой окно для доступа прикладных процессов к сетевым услугам. Этот уровень обеспечивает услуги, напрямую поддерживающие приложения. В действительности – это набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким, как файлы, принтеры или гипертекстовые Web-страницы, а также организуют свою совместную работу, например, с помощью протокола электронной почты. Единица данных, которой оперирует прикладной уровень, обычно называется *сообщением* (message).

Нижележащие уровни поддерживают задачи, решаемые на прикладном уровне. Этот уровень управляет общим доступом к данным и обработкой ошибок.

Существует очень большое разнообразие служб прикладного уровня. В качестве примера приведем несколько наиболее распространенных реализаций файловых служб: NCP в операционной системе Novell NetWare, FTP и TFTP, входящие в стек TCP/IP.

Сетезависимые и сетенезависимые уровни. Функции всех уровней модели OSI можно разбить на две группы: функции, зависящие от конкретной технической реализации сети и функции, ориентированные на работу с приложениями.

Три нижних уровня – физический, канальный и сетевой – являются сетезависимыми, то есть протоколы этих уровней тесно связаны с технической реализацией сети и используемым коммуникационным оборудованием. Например, переход на оборудование FDDI означает полную смену протоколов физического и канального уровней во всех узлах сети.

Три верхних уровня – прикладной, представительный и сеансовый – ориентированы на приложения и мало зависят от технических особенностей построения сети. На протоколы этих уровней не влияют какие бы то ни было изменения в топологии сети, замена оборудования или переход на другую сетевую технологию. Так, переход от Ethernet на высокоскоростную технологию 100VG-AnyLAN не потребует никаких изменений в программных средствах, реализующих функции прикладного, представительного и сеансового уровней.

Транспортный уровень является промежуточным, он скрывает все детали функционирования нижних уровней от верхних. Это позволяет разрабатывать приложения, не зависящие от технических средств непосредственной транспортировки сообщений.

Профили стандартов открытых систем

Идеи, заложенные в модели OSI, получили широкое международное признание. В различных странах и регионах создаются организации, которые проводят работы, связанные с созданием сетей, реализующих эту модель. Примерами таких учреждений являются ассоциация европейских производителей компьютеров, организация OSIone.

Поставщики, производители и организации по стандартизации проводят работы по определению подмножеств стандартов взаимодействия открытых систем, предназначенных для конкретных нужд пользователей. Эти подмножества называют *функциональными профилями.*

Модель OSI представляет хотя и очень важную, но только одну из многих моделей коммуникаций. Эти модели и связанные с ними стеки протоколов могут отличаться количеством уровней, их функциями, форматами сообщений, службами, поддерживаемыми на верхних уровнях, и прочими параметрами.

Интеграция компонентов в открытой системе должна следовать профилям стандартов на интерфейсы этих компонент.

Профили представляют собой набор согласованных стандартов интерфейсов компонентов на каждом уровне системы (как было показано выше на примере системы обработки данных) и обеспечивают их совместимость.

Для определенности рассмотрения интерфейсов компонент и проведения необходимых анализов их реализуемости можно использовать модель среды открытых систем MUSIC, разработанную центральным агентством по компьютерам и телекоммуникациям (CCTA) Великобритании. Эта модель является базовой фирмы Digital Equipment для построения открытых систем. Модель MUSIC включает пять групп компонентов, из которых строятся открытые системы:

- управление (Management) – функции системной администрации, безопасности, управления ресурсами;
- конфигурация, сетевое управление;
- пользовательский интерфейс (User Interface) – интерфейс пользователя с прикладными программами и со средой разработки приложений;
- системные интерфейсы для программ (Service Interface for Programs) – интерфейсы между прикладными программами и между прикладными программами и ОС, в частности API (Application Programs Interface);
- форматы информации и данных;
- интерфейсы коммуникаций.

Европейская рабочая группа по открытым системам (EWOS) предложила шесть профилей стандартов:

среда рабочих станций;
среда серверов процессов;
среда серверов данных;
среда транзакций;
среда реального времени;
среда суперкомпьютеров.

Кроме указанного набора профилей по классам аппаратно-программных средств существует необходимость формирования вертикальных профилей открытых систем, ориентированных на проблемно-ориентированные области применения. В качестве таких первоочередных областей применения открытых систем в России можно назвать:

- интегрированные производственные системы,
- информационные системы (системы информационного обслуживания) с удаленным доступом к ресурсам,
- системы автоматизации учреждений,
- системы автоматизации банков,
- системы автоматизации научных исследований,
- системы передачи данных.

2. ОСНОВЫ ТЕЛЕКОММУНИКАЦИИ

Даны основные понятия о системе передачи данных, каналах связи и их основных характеристиках, о формах представления сигналов в дискретном и непрерывном каналах связи Значительное внимание уделено методам защиты информации и сжатия при ее передаче по каналам связи Рассмотрены основные методы и технологии передачи данных на канальном уровне, в том числе по телефонным каналам связи В качестве примера технических средств телекоммуникаций рассмотрены основные связные интерфейсы и модемы , их характеристики и параметры

2.1. Понятие системы передачи данных

Для передачи информации используют некоторый материальный носитель – сигнал. Различают статические и динамические сигналы. *Статические сигналы* в основном предназначены для передачи информации во времени, т. е. для хранения информации с последующим ее использованием, *динамические сигналы* – для передачи информации в пространстве. Любой сигнал неразрывно связан с определенной материальной системой, называемой *системой связи* или *системой передачи информации* (рис. 2.1).

Будем считать, что с *источником* информации связано определенное множество сообщений. Генерация некоторого сообщения заключается в случайном выборе одного сообщения из множества возможных. Какое это конкретно

Рис. 2.1. Система передачи информации

будет сообщение, заранее неизвестно, по крайней мере тому, для кого оно предназначается. Известно лишь, что сообщение принадлежит определенному множеству.

Множества возможных сообщений бывают различных типов:

- конечные множества символов;
- конечные наборы детерминированных функций времени;
- бесконечные множества значений некоторой непрерывной физической величины.

Сообщение, принадлежащее конечному множеству возможных значений, называется *дискретным*, а сообщение, выбираемое из бесконечного множества – *непрерывным*.

Передатчик преобразует сообщение в передаваемый сигнал. В передатчике каждое из возможных сообщений на входе преобразуется в одно из возможных значений сигнала на выходе по строго определенному правилу. Правила, по которым осуществляется преобразование сообщения в сигнал, разные в зависимости от типов сообщений и сигналов (модуляция, кодирование, манипуляция).

Линия связи – собственно *физическая среда* (medium), по которой передаются сигналы. Одна и та же линия связи может служить одновременно для реализации одного или нескольких каналов связи (многоканальная связь).

Канал (канал связи) – средства односторонней передачи данных. Примером канала может служить полоса частот, выделенная одному передатчику при радиосвязи. В некоторой линии можно образовать несколько каналов связи, по каждому из которых передается своя информация. При этом говорят, что линия разделяется между несколькими каналами. Существуют два метода разделения линии передачи данных:

временное мультиплексирование (иначе разделение по времени или TDM – Time Division Method), при котором каждому каналу выделяется некоторый квант времени, *частотное разделение* (FDM – Frequency Division Method), при котором каналу выделяется некоторая полоса частот.

Принимаемый сигнал на выходе канала связи отличается от входного передаваемого сигнала из-за наложения *помехи* на полезный сигнал. *Приемник* осуществляет восстановление переданного источником информации сообщения по принятому сигналу. Данная операция возможна, если известно правило преобразования сообщения в сигнал. На основании этого правила вырабатывается правило обратного преобразования сигнала в сообщение (демодуляция, декодирование). Это правило позволяет в конечном счете выбрать приемной стороне сообщение из известного множества возможных сообщений, в идеальном случае полностью совпадающее с переданным сообщением.

Однако это бывает не всегда, вследствие искажения принятого сигнала возможна *ошибка* при восстановлении сообщения.

Получатель в системах передачи информации – это либо непосредственно человек, либо технические средства, связанные с человеком.

Типы линий связи

Физическая среда передачи данных может представлять собой кабель, т. е. набор проводов, изоляционных и защитных оболочек и соединительных разъемов, а также земную атмосферу или космическое пространство, через которые распространяются электромагнитные волны.

В зависимости от среды передачи данных различают следующие линии связи:

- проводные (воздушные);
- кабельные (медные и волоконно-оптические);
- радиоканалы наземной и спутниковой связи;
- инфракрасные лучи.

Проводные (воздушные) линии связи представляют собой провода без каких-либо изолирующих или экранирующих оплеток, проложенные между столбами и висящие в воздухе. По таким линиям связи традиционно передают телефонные или телеграфные сигналы, но при отсутствии других возможностей эти линии используют и для передачи компьютерных данных. Скоростные качества и помехозащищенность этих линий оставляют желать лучшего. Сегодня проводные линии связи быстро вытесняются кабельными.

Кабельные линии представляют собой достаточно сложную конструкцию. Кабель состоит из проводников, заключенных в несколько слоев изоляции: электрической, электромагнитной, механической, а также, возможно, климатической. Кроме того, кабель может быть оснащен разъемами, позволяющими быстро выполнять присоединение к нему различного оборудования. В системах телекоммуникации и компьютерных сетях применяют три основных типа кабеля: кабели на основе скрученных пар медных проводов, коаксиальные кабели с медной жилой, волоконно-оптические кабели.

Скрученная пара проводов называется *витой парой* (twisted pair). Витая пара изготовляется в двух вариантах: в экранированном (STP – Shiel-ded Twisted Pair) – когда пара медных проводов обертывается в изоляционный экран, и неэкранированном (UTP – Unshielded Twisted Pair) – когда изоляционная обертка каждой пары отсутствует. Скручивание проводов снижает влияние внешних помех на полезные сигналы, передаваемые по кабелю. *Коаксиальный кабель* (coaxial) имеет несимметричную конструкцию и состоит из внутренней медной жилы и оплетки, отделенной от жилы слоем изоляции. Существует несколько типов коаксиального кабеля, отличающихся характеристиками и областями применения – для локальных сетей, для глобальных сетей, для кабельного телевидения и т. п. *Волоконно-оптический кабель* (optical fiber) состоит из тонких (5...60 микрон) волокон, 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.

Радиоканалы наземной и спутниковой связи образуются с помощью передатчика и приемника радиоволн. Существует много типов радиоканалов, отличающихся как используемым частотным диапазоном, так и дальностью связи. Диапазоны коротких, средних и длинных волн (КВ, СВ и ДВ), называемые также диапазонами амплитудной модуляции (АМ – Amplitude Modulation) по типу используемого в них метода модуляции сигнала, обеспечивают дальнюю связь, но при невысокой скорости передачи данных. Более скоростными являются каналы, работающие на диапазонах ультракоротких волн (УКВ), для которых характерна частотная модуляция (FM – Frequency Modulation), а также на диапазонах сверхвысоких частот (СВЧ или microwaves). В диапазоне СВЧ (свыше 4 ГГц) сигналы уже не отражаются ионосферой Земли, и для устойчивой связи необходимо наличие прямой видимости между передатчиком и приемником. Поэтому такие частоты используют либо спутниковые каналы, либо радиорелейные каналы, где это условие выполняется.

Инфракрасное излучение. Инфракрасные беспроводные сети используют для передачи данных инфракрасные лучи. В подобных системах необходимо генерировать очень сильный сигнал, так как в противном случае значительное влияние будут оказывать другие источники.

Сети на рассеянном инфракрасном излучении. При этой технологии сигналы, отражаясь от стен и потолка, в конце концов достигают приемника. Эффективная область ограничивается примерно 30 м. Скорость передачи невелика (так как все сигналы отраженные).

Сети на отраженном инфракрасном излучении. В таких сетях оптические трансиверы, расположенные рядом с компьютером, передают сигналы в определенное место, из которого они транслируются соответствующему компьютеру.

Широкополосные оптические сети. Эти инфракрасные беспроводные сети предоставляют широкополосные услуги магистрали, соответствуют жестким требованиям мультимедийной среды и практически не уступают кабельным сетям. Хотя скорость и удобство использования инфракрасных сетей очень притягательны, возникают трудности при передаче сигналов на расстояние более 10 м. К тому же такие сети подвержены помехам со стороны сильных источников света, которые есть в большинстве помещений.

В компьютерных сетях в настоящее время применяют практически все описанные типы физических сред передачи данных, но наиболее перспективными являются волоконно-оптические. На них сегодня строят как магистрали крупных территориальных сетей, так и высокоскоростные линии связи локальных сетей. Популярной средой является также витая пара, которая характеризуется отличным соотношением качества к стоимости и простотой монтажа. С помощью витой пары обычно подключают конечных абонентов сетей на расстояниях до 100 м от концентратора. Спутниковые каналы и радиосвязь используют чаще всего в случаях, когда кабельные связи применить нельзя – например, при прохождении канала через малонаселенную местность или же для связи с мобильным пользователем сети.

Математические модели сигналов

Для передачи информации в качестве сигналов используют различные физические процессы или объекты, характеризующиеся большим числом параметров. Однако не все параметры этих процессов существенны с точки зрения передачи информации. Поэтому часто применяют приближенное представление физического процесса, используемого для передачи информации – *модель сигнала*.

Различают следующие параметры сигнала: структурные, идентифицирующие, информативные.

Структурные параметры определяют число степеней свободы сигнала. *Идентифицирующие* параметры служат для выделения полезного сигнала среди других сигналов, не предназначенных для данного адресата. *Информативные* используют для кодирования передаваемой информации.

Пример. Пусть математическое описание сигнала задано выражением:

$$S=X\sin(2\pi ft+\varphi) \qquad (2.1)$$

и возможные сообщения, выбираемые из множества C источником, преобразуются в передатчике в различные значения амплитуды X синусоидального колебания.

В этом случае амплитуда сигнала X является информативным параметром сигнала. По частоте f сигнала S обычно его выделяют среди других сигналов того же класса с другими значениями частоты. Таким образом, параметр f можно отнести к идентифицирующим параметрам. Число степеней свободы по информативному параметру сигнала S в общем случае зависит от времени – параметра t, поэтому t следует рассматривать как структурный параметр сигнала.

В случае, если информативный параметр X не зависит от структурного параметра t, то выбранное значение амплитуды остается неизменным на всем протяжении сигнала, т.е. каждое возможное сообщение сопоставляется с гармоническим колебанием бесконечной длительности и определенной амплитуды. Таким образом, в этом случае сигнал S по информативному параметру X имеет всего лишь одну степень свободы.

Если X зависит от параметра t в выражении $S(t)=X(t)\sin(2\pi ft+\varphi)$, то сигнал $S(t)$ в принципе имеет бесконечное число степеней свободы.

В качестве информативных можно использовать различные параметры, например f или φ, причем f может быть одновременно и информативным, и идентифицирующим параметром.

По информативным параметрам различают сигналы *дискретные* и *непрерывные*. Если множество возможных значений информативного параметра сигнала конечно или счетно, то сигнал называется *дискретным* по данному параметру. Если информативный параметр сигнала принимает континиум значений, то сигнал называется *непрерывным* по данному параметру.

Если информативный параметр не один, то сигнал может быть дискретным по одному параметру и непрерывным по другому. Поэтому часто бывает удобно пользоваться понятием *«состояние сигнала»*, которое определяется тем, какие конкретные значения примут k информативных параметров по каждой степени свободы.

Число возможных состояний сигнала

$$N = (m_1 m_2 \ldots m_i \ldots m_k)^n, \qquad (2.2)$$

где m_i – число возможных значений i-го параметра сигнала; n – число степеней свободы сигнала.

Из выражения (2.2) ясно, что если число степеней свободы сигнала или, по крайней мере, один из сомножителей бесконечно большой, то и число состояний сигнала также будет бесконечно большим. Так как в передатчике происходит изменение значений информативных параметров сигнала и, следовательно, изменение состояния сигнала в соответствии с передаваемым сообщением, то информация, переносимая сигналом, заключается именно в его состоянии.

Таким образом для любой модели сигнала (дискретные значения или непрерывные процессы) сущность процесса передачи информации не меняется и состоит в следующем:

- в передатчике сообщения трансформируются в состояние сигнала;
- сигнал в канале искажается помехой, и состояние сигнала непредсказуемо изменяется;
- в приемнике по измененному состоянию сигнала принимается решение относительно переданного сообщения.

Отсюда ясно, что при восстановлении сообщения возможны ошибки, и очевидно, что вероятность возникновения ошибок будет тем меньше, чем существенней в некотором смысле различаются между собой состояния сигнала, кодирующие различные сообщения. Следовательно, для того чтобы с помощью математической модели сигнала исследовать помехоустойчивость, в ней должна быть определена степень различия между возможными состояниями сигнала. Одним из приемов, позволяющим делать это, является представление возможных состояний сигнала в виде точек в некотором абстрактном пространстве, в котором тем или иным способом определено расстояние между любыми двумя точками, т. е. метрическое пространство. Как правило, в качестве модели сигнала используется *метрическое линейное пространство*, которое называют *пространством сигнала.*

В пространстве сигнала точек должно быть не меньше, чем возможных сообщений источника информации: $M \geq C$, где M – мощность множества пространства сигнала X, C – мощность множества сообщений источника. Сигналы, представляемые в пространствах, где $M = C$, обладают низкой устойчивостью к помехам.

Для повышения помехоустойчивости процесса передачи информации используют сигналы с бóльшим числом состояний, чем это необходимо для кодирования всех возможных сообщений, т. е. $M > C$. Тогда возникает вопрос: какие точки пространства сигнала сопоставлять возможным сообщениям источника информации? Чтобы ответить на этот вопрос, проведем анализ работы приемника.

На вход приемника поступает сигнал, искаженный помехой, которому соответствует точка x в пространстве сигнала X, отличная от той, которая была сопоставлена в передатчике передаваемому сообщению. Таким образом, в приемнике одному и тому же переданному сообщению могут соответствовать различные точки пространства сигнала. Чтобы приемник мог принимать каждый раз решение относительно переданного сообщения, пространство сигнала должно быть классифицировано, т. е. множество X должно быть априорно разбито на непересекающиеся подмножества (классы) $C_1, C_2, \ldots, C_m$ и установлено взаимно однозначное отображение разбиения $\{C_1, C_2, \ldots, C_m\}$ на множество возможных сообщений источника информации:

$$\{C_1, C_2, \ldots, C_m\} \leftrightarrow C.$$

Рассмотрим один из способов такого разбиения, основанный на выделении в пространстве сигнала так называемых реперных точек $x_1, x_2, \ldots, x_m$, которые являются представителями соответствующих классов $C_1, C_2, \ldots, C_m$.

В передатчике каждому передаваемому сообщению сопоставляется определенная реперная точка пространства сигнала. В процессе передачи помеха переводит эту реперную точку в другую точку x пространства сигнала. В приемнике осуществляется процесс, который, в сущности, сводится к оценке расстояния между точкой x пространства сигнала X и всеми реперными точками $x_1, x_2, \ldots, x_m$ и выбору той реперной точки, до которой от точки x расстояние минимально, т. е. вычисляется

$$\min_i d(x_i, x_j) \text{ для всех } i \text{ и } j \text{ от 1 до } m. \qquad (2.3)$$

Фактически в приемнике осуществляется классификация пространства сигнала, объединением точек, ближайших к данной реперной точке x_i в один класс $C_i(x, x_i)$ (рис. 2.2):

$$C_i(x, x_i) = \{x \in X | d(x, x_i) < d(x, x_j), i \neq j\}. \qquad (2.4)$$

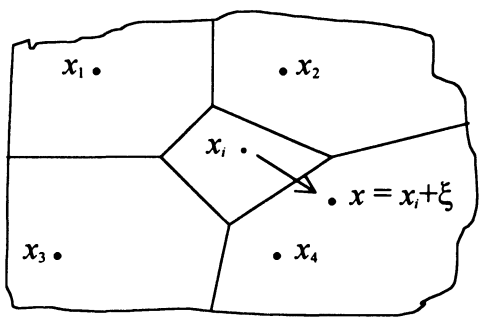


Рис. 2.2. Классификация пространства сигнала

На рис. 2.2 линии состоят из точек пространства, не вошедших ни в один из классов.

Искажение сигнала в канале можно рассматривать как наложение на выбранную передатчиком реперную точку x_i некоторой помехи ξ. В результате становится доступной для анализа в приемнике точка $x = x_i + \xi$. Значения x_i и ξ неизвестны. Поэтому возникает задача так распределить реперные точки при заданном статистическом описании сообщений и помехи, чтобы выход точки $x = x_i + \xi$ за границы класса $C_i(x, x_i)$ происходил бы как можно реже.

Рассмотрим одно из частных решений поставленной задачи. Пусть пространство сигнала есть линейное пространство, заданное в некотором ортонормированном базисе $u_1, u_2, \ldots, u_n$. В ортонормированном линейном векторном пространстве *норма* произвольного вектора

$$v = \sum_{i=1}^{n} \alpha_i u_i = \alpha_1 u_1 + \alpha_2 u_2 + \ldots + \alpha_k u_k + \ldots + \alpha_n u_n \qquad (2.5)$$

определится как

$$\| v \| = \sqrt{(v, v)} = \sqrt{\sum_{i=1}^{n} \alpha_i^2}, \qquad (2.6)$$

а расстояние между парой векторов v и w определяется выражением:

$$d(v, w) = \| v - w \| = \sqrt{\sum_{i=1}^{n} (\alpha_i - \beta_i)^2} = \sqrt{(v - w, v - w)}. \qquad (2.7)$$

Точка $x = \alpha_1 u_1 + \alpha_2 u_2 + \ldots + \alpha_k u_k + \ldots\ldots + \alpha_n u_n$ – произвольная точка в этом пространстве. Она рассматривается как возможное состояние сигнала, в частности возможное значение его информативного параметра.

В теории связи квадрат нормы вектора x обычно называют *энергией* сигнала.

$$E = ||x||^2 = (x, x). \qquad (2.8)$$

Предположим, что в рассматриваемом пространстве сигнала необходимо разместить m реперных точек $-x_1, x_2, \ldots, x_m$, расстояние между любой парой точек x_i и x_j:

$$d(x_i, x_j) = || x_i - x_j ||. \qquad (2.9)$$

Для уменьшения числа ошибок при восстановлении сообщений необходимо стремиться увеличивать расстояние между реперными точками.

Пусть E_i – энергия сигнала x_i, а E_j – энергия сигнала x_j. Умножим левую и правую части равенства (2.9) на вещественное неотрицательное число λ:

$$\lambda d(x_i, x_j) = || \lambda x_i - \lambda x_j ||. \qquad (2.10)$$

Испольуя выражение для энергии, для сигналов λx_i и λx_j получим значения энергий $\lambda^2 E_i$ и $\lambda^2 E_j$ соответственно. Отсюда следует, что при пропорциональ-

ном увеличении энергии сигналов x_i и x_j расстояние между ними увеличивается. Реальная энергия сигнала всегда ограничена. Поэтому будем решать задачу оптимального распределения реперных точек в пространстве сигнала при условии равенства конечных энергий сигналов $x_1, x_2, \ldots, x_m$, выполняющих роль реперных точек.

Из определения расстояния в линейных пространствах со скалярным произведением векторов, имеем:

$$d^2(x_i, x_j) = (x_i - x_j, x_i - x_j) = (x_i, x_i - x_j) - (x_j, x_i - x_j) = \\ = (x_i, x_i) - (x_i, x_j) - (x_j, x_i) + (x_j, x_j). \tag{2.11}$$

Учитывая, что $E_i = (x_i, x_i)$, а $E_j = (x_j, x_j)$ и по условию $E_i = E_j = E$ получим:

$$d^2(x_i, x_j) = 2E - 2(x_i, x_j), \tag{2.12}$$

т. е. расстояние между сигналами зависит не только от их энергии, но и от их скалярного произведения.

Учитывая, что $-E \le (x_i, x_j) \le E$, представим скалярное произведение (x_i, x_j) в виде произведения $\lambda_{ij}E$, где λ_{ij} – коэффициент различимости сигналов $(-1 \le \lambda_{ij} \le 1)$:

$$d^2(x_i, x_j) = 2E(1 - \lambda_{ij}). \tag{2.13}$$

Из формулы (2.13) видно, что расстояние между сигналами минимально и равно нулю, когда $x_i = x_j$, при этом $(x_i, x_j) = E$, $\lambda_{ij} = 1$, $d(x_i, x_j) = 0$.

Расстояние между сигналами x_i и x_j равной энергии максимально, когда $x_i = -x_j$. В этом случае $\lambda_{ij} = -1$, а $d^2(x_i, x_j) = 4E$.

Если в пространстве сигнала необходимо разместить только две реперные точки, то вопрос об их оптимальном распределении решается весьма просто: нужно выбрать произвольный сигнал x_1 заданной энергии и в качестве второго сигнала x_2 взять сигнал $-x_1$.

Количественная оценка информационного содержания сигнала

Рассмотрим дискретный по параметру информативности сигнал. С помощью этого сигнала можно закодировать конечное множество возможных сообщений. Интуитивно понятно, что количество информации, которое получает адресат, некоторым образом связано с априорной неопределенностью ситуации, зависящей, в конечном счете, от числа возможных сообщений. Таким образом, чем больше число возможных сообщений и, следовательно, чем больше возможных значений сигнала, тем больше априорная неопределенность и тем большее количество информации получает адресат, когда эта неопределенность снимается.

Впервые количественную оценку неопределенности ввел в 1928 г. Р. Хартли для опыта X с m различными исходами:

$$H(X) = \log m. \tag{2.14}$$

Под опытом X можно понимать информативный параметр сигнала.

Однако в оценке Р. Хартли не учтены вероятности различных исходов. К. Шеннон ограничил рамки применимости оценки Р. Хартли случаем, когда все m исходов в опыте X равновероятны ($p = 1/m$), а затем применил формулу к разновероятным исходам, усреднив полученные неопределенности по всем исходам.

Для опыта $X = \{x_1, x_2, \ldots, x_m\}$, где $x_1, x_2, \ldots, x_m$ – возможные исходы с вероятностями $p_1, p_2, \ldots, p_m$, неопределенность каждого исхода равна $-\log p_1$, $-\log p_2, \ldots, -\log p_m$, а математическое ожидание дает количественную оценку неопределенности – энтропию:

$$H(X) = -\sum_{i=1}^{m} p_i \log p_i. \tag{2.15}$$

Понятие энтропии тесно связано с понятием количества информации. Под количеством информации понимается мера снятия неопределенности в процессе получения сигнала адресатом.

Пример. Априорно ситуация характеризовалась энтропией H_1. После получения сигнала, энтропия уменьшилась до H_2. Количество информации, полученной адресатом, равно $I = H_1 - H_2$. Если неопределенность снята полностью ($H_2 = 0$), то $I = H_1$.

Рассмотрим свойства, которыми обладает энтропия дискретного сигнала. Энтропия заранее известного сигнала (значение его информативного параметра априорно известно) равна нулю. Формула для энтропии в этом случае будет состоять из слагаемых только двух видов: либо $1 \times \log 1$ для заранее известного сигнала, либо $0 \times \log 0$, так как вероятность появления всех других равна нулю.

Так как $1 \times \log 1 = 0$ и $\lim\limits_{x \to 0}(x \times \log x) = 0$, то энтропия заранее известного сигнала равна нулю.

Энтропия – вещественная и неотрицательная величина. Это справедливо, так как перед знаком суммы в формуле энтропии стоит знак минус, а вероятности неотрицательны и не превышают значение единицы. Энтропия – величина конечная при любом конечном числе m.

Продифференцируем и приравняем нулю производную:

$$\frac{d}{dp_i}(-p_i \log p_i) = -\log p_i - p_i \frac{1}{p_i} \log e = 0. \tag{2.16}$$

Отсюда следует, что $p_i = 1/e$ и все слагаемые в формуле для энтропии не превышают значение $1/e \times \log e$. Следовательно, $H(X)$ – конечна.

Энтропия достигает максимального значения, когда вероятности появления возможных значений информативного параметра сигнала одинаковы.

Найдем максимум функции

$$F = -\sum_{i=1}^{m} p_i \log p_i - \lambda \sum_{i=1}^{m} p_i \qquad (2.17)$$

методом неопределенных множителей Лагранжа при дополнительном условии

$$\sum_{i=1}^{m} p_i = 1.$$

Дифференцировав F по p_i и приравняв производную нулю, получим:

$$-\log p_i - 1/p_i\, p_i \log e - \lambda = 0$$

или

$$-\log p_i = \log e + \lambda, \qquad (2.18)$$

т. е. вероятность p_i не зависит от переменной суммирования i. Это может быть лишь в том случае, когда все вероятности равны между собой: $p_1 = p_2 = \ldots = p_m = p = 1/m$.

Следовательно,

$$H_{\max} = -\sum_{i=1}^{m} \frac{1}{m} \log\left(\frac{1}{m}\right) = \log m, \qquad (2.19)$$

таким образом энтропия достигает своего максимального значения при равновероятных значениях информативного параметра сигнала и равна оценке Р. Хартли.

Для получения количественной оценки энтропии обычно используют основание логарифма равное двум, при этом полученная единица измерения количества информации называется битом (bit – binary digit).

Непрерывный и дискретный каналы

В зависимости от того, какие сигналы передаются по каналу связи, различают аналоговые (непрерывные) и цифровые (дискретные) каналы.

В аналоговых каналах передатчик (см. рис. 2.1) выполняет роль устройства согласования источника сообщений с непрерывным каналом, т.е. осуществляет преобразование непрерывного или дискретного сообщения в непрерывный по структурному параметру сигнал с такими характеристиками, которые обеспечивают его прохождение по данному каналу связи. В таких каналах для согласования параметров среды и сигналов применяют амплитудную, частотную, фазовую и квадратурно-амплитудную модуляции.

В цифровых каналах на выходе передатчика и входе приемника действует дискретный по структурному параметру сигнал. В них для передачи данных используют самосинхронизирующиеся коды, а для передачи аналоговых сигналов – кодово-импульсную модуляцию.

Обычно дискретным каналом называют комплекс технических средств, обеспечивающих передачу дискретного сигнала. Во многих системах переда-

чи данных дискретный канал включает непрерывный канал связи. Однако при анализе дискретного канала свойства непрерывного канала учитывают косвенно через свойства источника ошибок.

Основными характеристиками непрерывных каналов связи являются:
амплитудно-частотная характеристика,
полоса пропускания,
затухание,
помехоустойчивость,
шумы,
пропускная способность,
достоверность передачи данных,
удельная стоимость.

Разработчика вычислительной сети в первую очередь интересуют пропускная способность и достоверность передачи данных, поскольку эти характеристики прямо влияют на производительность и надежность создаваемой сети. Пропускная способность и достоверность передачи данных – это характеристики как канала связи, так и способа передачи данных. Поэтому если способ передачи (протокол) уже определен, то известны и эти характеристики. Например, пропускная способность цифрового канала всегда известна, так как на нем определен протокол физического уровня, который задает битовую скорость передачи данных – 64 кбит/с, 2 Мбит/с и т. п.

Однако нельзя говорить о пропускной способности канала связи, до того как для него определен протокол физического уровня. Именно в таких случаях, когда только предстоит определить, какой из множества существующих протоколов можно использовать для данного канала, очень важными являются остальные характеристики, такие как полоса пропускания, перекрестные наводки, помехоустойчивость и др.

Для определения характеристик канала связи часто используют анализ его реакций на некоторые эталонные воздействия. Такой подход позволяет достаточно просто и однотипно определять характеристики линий связи любой природы, не прибегая к сложным теоретическим исследованиям. Чаще всего в качестве эталонных сигналов для исследования реакций линий связи используют синусоидальные сигналы различных частот. Это связано с тем, что сигналы этого типа часто встречаются в технике и с их помощью можно представить любую функцию времени – как непрерывный процесс колебаний звука, так и прямоугольные импульсы, генерируемые компьютером.

Из теории гармонического анализа известно, что любой периодический процесс можно представить в виде суммы синусоидальных колебаний различных частот и амплитуд (рис. 2.3). Каждая составляющая синусоида называется гармоникой, а набор всех гармоник – спектральным разложением исходного сигнала. Непериодические сигналы можно представить в виде интеграла сину-

соидальных сигналов с непрерывным спектром частот. Например, спектральное разложение идеального импульса (единичной мощности и нулевой длительности) имеет составляющие всего спектра частот, от – ∞ до + ∞ (рис. 2.4).

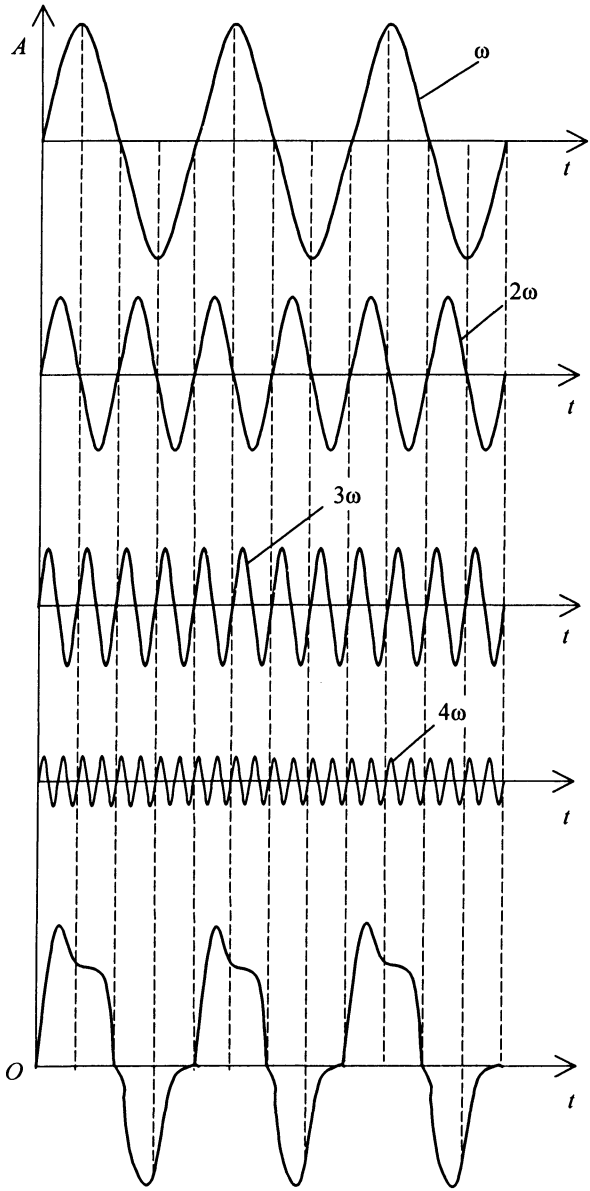


Рис. 2.3. Представление периодического сигнала суммой синусоид

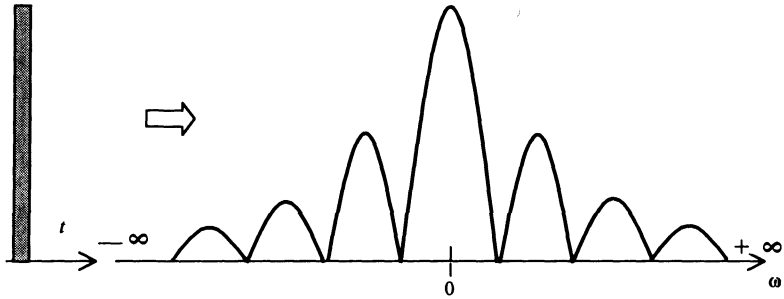


Рис. 2.4. Спектральное разложение идеального импульса

Техника определения спектра любого исходного сигнала известна. Для некоторых сигналов, которые хорошо описываются аналитически (например, для последовательности прямоугольных импульсов одинаковой длительности и амплитуды), спектр легко вычисляется по формулам 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 или распечатывают их на принтере.

Искажение передающим каналом синусоиды какой-либо частоты приводит в конечном счете к искажению передаваемого сигнала любой формы, особенно если 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. Воздействия на сигнал различных факторов в процессе передачи изображены на рис. 2.5. Как видно, сигнал, передаваемый по любой среде передачи, подвергается воздействию затухания, ограниченности полосы пропускания, задержки передачи и шумов. Хотя все эти факторы оказывают совокупное воздействие, рассмотрим источник каждого из них в отдельности.

Затухание (attenuation) определяется как относительное уменьшение амплитуды или мощности сигнала при передаче по каналу сигнала определенной частоты. Таким образом, затухание представляет собой одну точку на амплитудно-частотной характеристике. Амплитудно-частотная характеристика (рис. 2.6) показывает, как затухает амплитуда синусоиды (или мощность) на выходе канала связи по сравнению с амплитудой (или мощностью) на его входе для всех возможных частот передаваемого сигнала.

Часто при эксплуатации канала заранее известна основная частота передаваемого сигнала, т. е. та частота, гармоника которой имеет наибольшую амплитуду и мощность. Поэтому достаточно знать затухание на этой частоте, чтобы приблизительно оценить искажения передаваемых по каналу сигналов. Более точные оценки возможны при известном затухании на нескольких частотах, соответствующих нескольким основным гармоникам передаваемого сигнала.

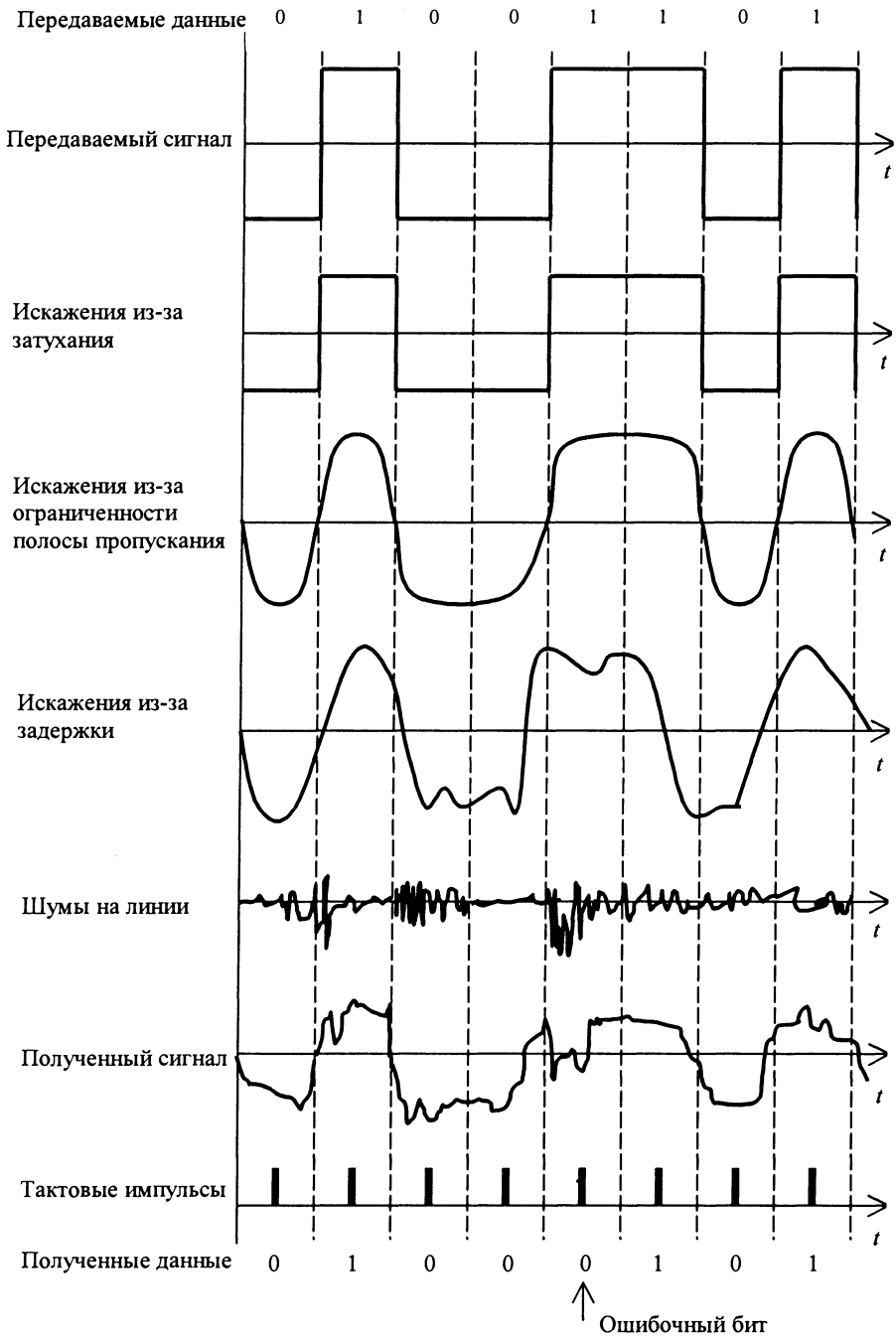


Рис. 2.5. Источники затухания и искажения сигнала

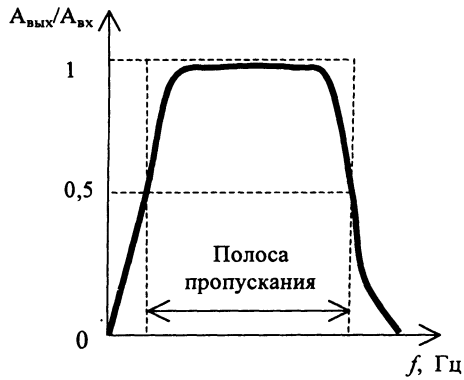


Рис. 2.6. Амплитудно-частотная характеристика

Затухание A (дБ) вычисляют по формуле:

$$A = 10 \lg P_{вых}/P_{вх}, \qquad (2.20)$$

где $P_{вых}$ – мощность сигнала на выходе канала, $P_{вх}$ – мощность сигнала на входе канала.

Так как мощность выходного сигнала среды передачи без промежуточных усилителей всегда меньше, чем мощность входного сигнала, затухание среды передачи всегда является отрицательной величиной.

Например, витая пара – кабель категории 5 – характеризуется затуханием не ниже – 23,6 дБ для частоты 100 МГц при длине кабеля 100 м. Частота 100 МГц выбрана потому, что кабель этой категории предназначен для высокоскоростной передачи данных, сигналы которых имеют значимые гармоники с частотой примерно 100 МГц. Кабель категории 3 предназначен для низкоскоростной передачи данных, поэтому для него определяется затухание на частоте 10 МГц (не ниже –11,5 дБ). Часто оперируют с абсолютными значениями затухания, без указания знака.

Абсолютный уровень мощности, например, уровень мощности передатчика, также измеряется в децибелах. При этом в качестве базового значения мощности сигнала, относительно которого измеряется текущая мощность, принимается значение в 1 мВт. Таким образом, уровень мощности, дБм,

$$p = 10 \lg P/1 \text{ мВт}, \qquad (2.21)$$

где P – мощность сигнала, мВт; дБм (*dBm*) – единица измерения уровня мощности (децибел на 1 мВт).

Полоса пропускания (bandwidth) – непрерывный диапазон частот, для которого отношение амплитуды выходного сигнала ко входному превышает некоторый заранее заданный предел, обычно 0,5; т. е. полоса пропускания определяет диапазон частот синусоидального сигнала, при которых этот сигнал передается по каналу связи без значительных искажений.

Так как любой дискретный сигнал состоит из компонент различной частоты, то на вход приемного устройства поступают только те компоненты, частоты которых находятся внутри полосы пропускания. Ограниченность полосы пропускания приводит к частотным искажениям сигнала. Известно, что амплитуда каждой из частотных гармоник снижается с ростом частоты. Поэтому, чем шире полоса пропускания среды передачи, тем большее число высокочастотных компонент проходит по линии связи, а следовательно, тем надежнее будет полученный сигнал воспроизводить переданный сигнал.

Искажение из-за задержки определяется тем, что скорость распространения синусоидального сигнала по линии связи изменяется с изменением частоты. Следовательно, при передаче цифрового сигнала различные компоненты, из которых образован сигнал, достигают приемника с различными задержками. Результатом этого является искажение сигнала, называемое искажением, вызванным задержкой. Степень искажения растет с увеличением скорости передачи битов, что вызвано следующей причиной: по мере роста скорости битов некоторые частотные компоненты, связанные с передачей данного бита, задерживаются и начинают влиять на частотные компоненты следующего бита. Поэтому искажения из-за задержки называют также межсимвольными взаимными помехами. В результате действия этого искажения в моменты измерения поступивший сигнал изменяется. Так как обычно поступивший сигнал измеряется в номинальном центре каждого битового интервала, то, следовательно, при увеличении скорости битов искажение из-за задержки может привести к некорректной интерпретации полученного сигнала.

Шумы постоянно присутствуют в реальном канале. В отсутствие передаваемого сигнала в идеальной линии связи должен быть нулевой уровень электрического сигнала. Однако на практике в линии имеют место случайные всплески даже тогда, когда никакой сигнал не передается. Эти всплески называют уровнем шумов в линии, и в пределе по мере затухания передаваемого сигнала его уровень становится сравнимым с уровнем шума. Важным параметром, связанным со средой передачи является отношение мощности полученного сигнала P_S к мощности уровня шумов P_N: $SNP = P_S / P_N$. Отношение S/N называют отношением сигнал–шум и обычно выражают в децибелах:

$$S/N = 10 \lg (SNP). \qquad (2.22)$$

Совершенно очевидно, что высокое значение отношения свидетельствует о высокой мощности сигнала по отношению к имеющемуся уровню шумов и поэтому характеризует сигнал хорошего качества. Наоборот, низкое значение отношения S/N свидетельствует о сигнале низкого качества.

Помехоустойчивость линии определяет ее способность уменьшать уровень помех, создаваемых внешней средой, на внутренних проводниках. Помехоустойчивость линии зависит от типа используемой физической среды, а также от экранирующих и подавляющих помехи средств самой линии. Наименее помехоустойчивыми являются радиолинии, хорошей устойчивостью обладают

кабельные линии и отличной – волоконно-оптические линии, малочувствительные ко внешнему электромагнитному излучению. Обычно для уменьшения помех, появляющихся из-за внешних электромагнитных полей, проводники экранируют и/или скручивают.

Перекрестные наводки NEXT (Near End Cross Talk) определяют помехоустойчивость кабеля к внутренним источникам помех, когда электромагнитное поле сигнала, передаваемого выходом передатчика по одной паре проводников, наводит на другую пару проводников сигнал помехи. Если ко второй паре будет подключен приемник, то он может принять наведенную внутреннюю помеху за полезный сигнал. Показатель NEXT, выраженный в децибелах, равен

$$\text{NEXT} = 10 \lg P_{\text{вых}}/P_{\text{нав}},$$

где $P_{\text{вых}}$ – мощность выходного сигнала, $P_{\text{нав}}$ – мощность наведенного сигнала.

Показатель NEXT обычно используют применительно к кабелю, состоящему из нескольких витых пар, так как в этом случае взаимные наводки одной пары на другую могут достигать значительных величин.

Пропускная способность (throughput) линии характеризует максимально возможную скорость передачи данных по линии связи. Пропускная способность измеряется в битах в секунду – бит/с, а также в производных единицах: кбит/с, Мбит/с и т. д.

Пропускная способность линии связи зависит не только от ее характеристик, таких как амплитудно-частотная характеристика, но и от того, какие сигналы передаются – аналоговые или цифровые. Если значимые гармоники сигнала (т. е. 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. Если же значимые гармоники выходят за границы полосы пропускания линии связи, то сигнал будет значительно искажаться, приемник будет ошибаться при распознавании информации, а значит, информация не сможет передаваться с заданной пропускной способностью.

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 характеризоваться одной пропускной способностью, а для другого – другой.

Теория информации говорит, что любое различимое и непредсказуемое изменение принимаемого сигнала несет в себе информацию. В соответствии с этим прием синусоиды, у которой амплитуда, фаза и частота остаются неизменными, информации не несет, так как изменение сигнала хотя и происходит, но является хорошо предсказуемым. Большинство способов кодирования используют изменение какого-либо параметра периодического сигнала: частоты,

амплитуды и фазы синусоиды или же знак потенциала последовательности импульсов. Периодический сигнал, параметры которого изменяются, называют несущим сигналом, сигналом-переносчиком или несущей частотой, если в качестве такого сигнала используется синусоида.

Если сигнал изменяется так, что равновероятно можно различить только два состояния его информативного параметра, то в соответствии с оценкой Р. Хартли любое изменение сигнала, как отмечалось выше, будет соответствовать наименьшей единице информации – биту. Если же сигнал может иметь более двух различимых состояний, то любое его изменение будет нести несколько бит информации.

Количество изменений информативного параметра несущего периодического сигнала в секунду измеряется в бодах (*baud*). Период времени между соседними изменениями информативного параметра сигнала называется тактом работы передатчика или бодовым интервалом.

Пропускная способность линии в бит/с в общем случае не совпадает с числом бод. Она может быть как выше, так и ниже числа бод, что зависит от способа кодирования.

Если сигнал имеет более двух различимых состояний, то пропускная способность в бит/с будет выше, чем число бод. Например, если информативными параметрами являются фаза и амплитуда синусоиды, причем различаются 4 состояния фазы в 0, 90, 180 и 270° и два значения амплитуды сигнала, то информационный сигнал может иметь 8 различимых состояний. В этом случае модем, работающий со скоростью 2400 бод (с тактовой частотой 2400 Гц) передает информацию со скоростью 7200 бит/с, так как при одном изменении сигнала передается 3 бита информации.

При использовании сигналов с двумя различимыми состояниями может наблюдаться обратная картина. Это часто происходит потому, что для надежного распознавания приемником пользовательской информации каждый бит в последовательности кодируется с помощью нескольких изменений информативного 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 пропускная способность линии в два раза ниже, чем число бод, передаваемое по линии.

На пропускную способность линии оказывает влияние не только физическое, но и логическое кодирование. Логическое кодирование выполняется до физического кодирования и подразумевает замену бит исходной информации новой последовательностью бит, несущей ту же информацию, но обладающей, кроме этого, дополнительными свойствами, например возможностью для приемной стороны обнаруживать ошибки в принятых данных. Сопровождение каждого байта исходной информации одним битом четности – это пример очень часто применяемого способа логического кодирования при передаче данных с помощью модемов. Другим примером логического кодирования может слу-

жить шифрация данных, обеспечивающая их конфиденциальность при передаче через общественные каналы связи. При логическом кодировании чаще всего исходная последовательность бит заменяется более длинной последовательностью, поэтому пропускная способность канала по отношению к полезной информации при этом уменьшается.

Достоверность передачи данных характеризует вероятность искажения для каждого передаваемого бита данных. Иногда этот же показатель называют интенсивностью битовых ошибок (BER – Bit Error Rate). Значение BER для каналов 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}. Значение достоверности передачи данных, например, -10^{-4} говорит о том, что в среднем из 10000 бит искажается значение одного бита.

Искажения бит происходят как из-за наличия помех на линии, так и по причине искажений формы сигнала ограниченной полосой пропускания линии. Поэтому для повышения достоверности передаваемых данных нужно повышать степень помехозащищенности линии, снижать уровень перекрестных наводок в кабеле, а также использовать более широкополосные линии связи.

Пропускная способность среды передачи

Чем выше частота несущего периодического сигнала, тем больше информации в единицу времени передается по линии и тем выше пропускная способность линии при фиксированном способе физического кодирования. Однако, с другой стороны, с увеличением частоты периодического несущего сигнала увеличивается и ширина спектра этого сигнала, т. е. разность между максимальной и минимальной частотами того набора синусоид, которые в сумме дадут выбранную для физического кодирования последовательность сигналов. Линия связи 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, скорость передачи информации на самом деле оказывается меньше, чем можно было предположить.

Связь между полосой пропускания линии и ее максимально возможной пропускной способностью, вне зависимости от принятого способа физического кодирования, установили Шеннон и Хартли. Эта формула называется законом Шеннона–Хартли:

$$C = B \log_2(1 + SNP), \qquad (2.23)$$

где C – максимальная пропускная способность линии, бит/с; B – ширина полосы пропускания линии, Гц; SNP – отношение мощностей сигнала и шума.

Из формулы (2.23) видно, что хотя теоретического предела пропускной способности линии с фиксированной полосой пропускания не существует, на практике такой предел имеется. Действительно, повысить пропускную способность линии можно за счет увеличения мощности передатчика или же уменьшения мощности шума (помех) на 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 так же 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.

Найквист вывел формулу, определяющую зависимость максимальной скорости передачи информации (данных) C [бит/с] от ширины полосы пропускания B без учета шума в канале:

$$C = 2B \log_2 M, \qquad (2.24)$$

где M – число различимых состояний информативного параметра сигнала.

Если сигнал имеет 2 состояния, то пропускная способность равна удвоенному значению ширины полосы пропускания линии связи.

Если же передатчик использует более чем 2 устойчивых состояния сигнала для кодирования данных, то пропускная способность линии повышается, так как за один такт работы передатчик передает несколько бит исходных данных.

Пример. Модем в телефонной сети общего пользования применяет метод квадратурной амплитудной модуляции с 8-ю уровнями (4 значения фазы х 2 значения амплитуды для каждой фазы) на каждый сигнальный элемент. Если полоса пропускания телефонной сети равна 3100 Гц, то согласно формуле Найквиста максимальная скорость передачи данных будет равна:

$$C = 2B \log_2 M = 2 \cdot 3100 \cdot \log_2 8 = 18600 \text{ бит/с}.$$

Хотя формула Найквиста явно не учитывает наличие шума, косвенно его влияние отражается в выборе числа состояний информативного параметра сигнала. Для повышения пропускной способности канала хотелось бы увеличить это число до значительной величины, но на практике этого сделать нельзя из-за шума на линии. Поэтому число 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.

Передача данных на физическом уровне

Под данными понимают информацию, закодированную в цифровой форме. При передаче данных по каналам связи применяют два основных типа физи-

ческого кодирования – на основе синусоидального несущего сигнала и на основе последовательности прямоугольных импульсов. Первый способ часто называют также модуляцией или аналоговой модуляцией, подчеркивая тот факт, что кодирование осуществляется за счет изменения параметров аналогового сигнала. Второй способ обычно называют цифровым кодированием. Эти способы отличаются шириной спектра результирующего сигнала и сложностью аппаратуры, необходимой для их реализации.

При использовании прямоугольных импульсов спектр результирующего сигнала получается весьма широким. Это не удивительно, если вспомнить, что спектр идеального импульса имеет бесконечную ширину. Применение синусоиды приводит к спектру гораздо меньшей ширины при той же скорости передачи информации. Однако для реализации синусоидальной модуляции необходима более сложная и дорогая аппаратура, чем для реализации прямоугольных импульсов.

В настоящее время все чаще данные, изначально имеющие аналоговую форму – речь, телевизионное изображение, – передают по каналам связи в дискретном виде, т. е. в виде последовательности единиц и нулей. Процесс представления аналоговой информации в дискретной форме называется *дискретной модуляцией.*

При передаче данных по непрерывному (аналоговому) каналу связи используют определенный физический процесс, называемый сигналом-переносчиком. Математической моделью его может служить функция времени $s(t, A, B, \ldots)$, зависящая также от параметров A, B, ... Некоторые параметры сигналов фиксированы при данных условиях передачи, и тогда они выполняют роль идентифицирующих параметров. Другие подвергаются воздействию со стороны передатчика, и в этом случае выполняют роль информативных параметров.

Модуляция – отображение на передающей стороне множества возможных значений входного сигнала на множество возможных значений информативного параметра сигнала-переносчика. На приемной стороне возникает обратная задача – восстановить исходный сигнал, т. е. осуществить демодуляцию.

Как правило, аналоговую модуляцию применяют для передачи дискретных данных по каналам с узкой полосой частот, типичным представителем которых является канал тональной частоты (ТЧ), предоставляемый в распоряжение пользователям общественных телефонных сетей. Этот канал передает частоты в диапазоне от 300 до 3400 Гц, таким образом, его полоса пропускания равна 3100 Гц. Строгое ограничение полосы пропускания канала ТЧ связано с использованием аппаратуры уплотнения и коммутации каналов в телефонных сетях. Устройство, осуществляющее модуляцию несущей синусоиды на передающей стороне и демодуляцию на приемной стороне, носит название модем (модулятор–демодулятор).

Амплитудная модуляция. В системах с амплитудной модуляцией (АМ) модулирующая функция $\lambda(t)$ изменяет амплитуду высокочастотной гармонической функции $s(t)$ сигнала-переносчика:

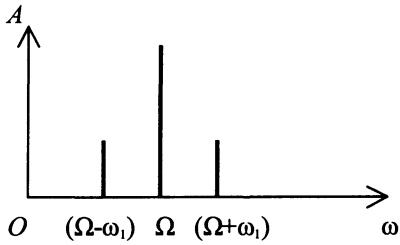


Рис. 2.7. Спектр амплитудно-модулированного сигнала

$$s(t) = A\sin(\Omega t + \Phi). \qquad (2.25)$$

Амплитудно-модулированный сигнал имеет вид:

$$s(t) = A[1 + m\lambda(t)]\sin(\Omega t + \Phi), \qquad (2.26)$$

где m – коэффициент модуляции.

Пусть модулирующая функция $\lambda(t) = \sin \omega_1 t$, тогда, подставив ее в выражение для $s(t)$ и осуществив преобразования, получим:

$$s(t) = A\{\sin(\Omega t + \Phi) + \frac{m}{2}\cos[(\Omega - \omega_1)t + \Phi] - \frac{m}{2}\cos[(\Omega + \omega_1)t + \Phi]\}. \qquad (2.27)$$

Амплитудно-модулированный сигнал имеет дискретный (линейчатый) спектр, состоящий из трех линий (рис. 2.7): несущей частоты – Ω и двух боковых частот $(\Omega - \omega_1)$ и $(\Omega + \omega_1)$ – одна ниже, другая выше несущей частоты. Их называют верхней и нижней боковыми частотами. Нижняя боковая – это зеркальное отображение верхней боковой по отношению к частоте несущей Ω. Из формулы (2.27) видно, что вся информация о модулирующей функции полностью содержится в любой из боковых частот.

Система с АМ, которая передает обе боковых и несущую частоту, известна, как двухполосная система (DSB – double sidebaud). Несущая не несет никакой полезной информации и может быть удалена, но с несущей или без, полоса сигнала DSB вдвое больше полосы изначального сигнала. Для сужения рабочей полосы частот канала связи возможно вытеснение не только несущей, но и одной из боковых, так как они несут одну информацию. Этот вид АМ известен, как однополосная модуляция с подавленной несущей SSB-SC (Single SideBand Suppressed Carrier). Этот вид модуляции создает новый сигнал, идентичный оригиналу, но сдвинутый вверх по частоте. Частоту несущей выбирают в соответствии с условиями среды передачи. Демодуляция сигнала АМ достигается путем смешивания модулированного сигнала с несущей той же самой частоты, что и на модуляторе. Изначальный сигнал затем получают как отдельную частоту (или полосу частот) и его можно отфильтровать от других сигналов. При использовании SSB-SC несущая для демодуляции генерируется на месте, и она может не совпадать с частотой несущей на модуляторе. Небольшая разница между двумя несущими частотами является причиной несовпадения восстанавливаемых частот, что присуще телефонным цепям.

Амплитудная модуляция с использованием цифровых сигналов. Особым случаем амплитудной модуляции является случай, когда нижний из двух уровней амплитуд доведен до нуля; тогда процесс модуляции состоит во включении и выключении несущей. Однако скачки в передаваемой энергии делают этот вид модуляции, не подходящим для передачи данных по сетям связи. Прямоугольная волна содержит высокочастотные компоненты, и на практике в

системах АМ-сигнал данных пропускают через фильтр нижних частот до модулятора. Это скругляет прямоугольную волну, но не влияет на информацию, содержащуюся в сигнале данных. Поскольку бинарный сигнал данных имеет составляющие вплоть до нулевой частоты, верхняя и нижняя боковые частоты фактически встретились на частоте Ω. Это обстоятельство делает затруднительным подавление несущей или одной боковой и несущей, без влияния на оставшуюся полосу. Для уменьшения полосы модулированного сигнала можно реально убрать бóльшую часть одной полосы, оставив только небольшой ее конец рядом с несущей. Потери информации нет, так как нижняя полоса просто дублирует информацию верхней полосы. Описанный подход называется VSB (VSB – vestigial sideband) – модуляция с частично подавленной боковой. При разумном построении фильтра в системах VSB можно подавить несущую. Это приведет к подавлению и части верхней полосы, но остаток нижней полосы, который будет сохранен, восполнит недостающие частоты. Правильная однополосная амплитудная модуляция с цифровым модулирующем сигналом может быть достигнута только путем скрэмблирования (scrambeing – перемешивание) изначальных данных (т. е. внося беспорядочность в поток бит) с целью удаления низкочастотных компонентов, которые образуются от ряда последовательных единиц или нулей. Это влечет рассоединение боковых частот от несущей частоты, что позволяет отфильтровать одну боковую и несущую.

Импульсная амплитудная модуляция (PAM – pulse amplitude modulation). Она использует модулирующий цифровой сигнал и реализует кодирование более чем одного бита на бод путем кодирования бинарного сигнала данных в сигнал с более чем двумя уровнями. Для примера, биты бинарного сигнала данных могут быть разбиты на пары. Возможны четыре комбинации пары бит и каждая пара может быть представлена одним из четырех уровней амплитуды. Закодированный четырехуровневый сигнал имеет половину скорости в бодах изначального сигнала данных и может быть использован для амплитудной модуляции несущей обычным образом.

Частотная модуляция. В системах частотной модуляции (ЧМ) частота несущей изменяется в соответствии с формой модулирующего сигнала. В этом случае частота Ω несущей (сигнала-переносчика $s(t) = A\sin \Omega t$) модулируется функцией $\cos \omega_1 t$:

$$\omega = \Omega[1 + m\lambda(t)] = \Omega[1 + (\Delta\omega / \Omega) \cos \omega_1 t], \qquad (2.28)$$

где $\Delta\omega / \Omega$ – коэффициент модуляции (относительное изменение частоты); $\Delta\omega$ – девиация частоты.

Тогда сигнал-переносчик

$$s(t) = A(\sin \int_0^t \omega dt = A(\sin\Omega t \cos(\beta \sin \omega_1 t) + \cos \Omega t \sin(\beta \sin \omega_1 t). \qquad (2.29)$$

Здесь $\beta = \Delta\omega/\omega_1$ – индекс модуляции.

При $\beta << 1$

$$s(t) \approx A(\sin \Omega t + \beta \sin \omega_1 t \cos \Omega t) = \\ = A[\sin \Omega t + (\beta/2) \sin(\omega_1 + \Omega) t + (\beta/2) \sin(\omega_1 - \Omega) t], \quad (2.30)$$

т. е. спектр частот ЧМ-сигнала практически не отличается от спектра АМ-сигнала.

Системы, в которых модулирующим сигналом является бинарный сигнал и, следовательно, несущая переключается сигналами с одной частоты на другую при неизменной амплитуде, называют системами FSK (Frequency Shift Keying)

Частотная модуляция помехоустойчива, поскольку искажению при помехах подвергается в основном амплитуда сигнала, а не частота. Необходимая для этого вида модуляции ширина спектра сигнала может быть значительно уже всей полосы пропускания канала. Частотная модуляция превосходит амплитудную в устойчивости к некоторым воздействиям, присутствующим в телефонной сети и ее следует использовать на более низких скоростях, где не требуется большая полоса частот. FSK является асинхронной техникой модуляции, для нее не требуется синхроимпульсов в модеме.

Фазовая модуляция. При фазовой модуляции (ФМ) информативным параметром сигнала-переносчика служит фаза Ф несущей частоты Ω:

$$s(t) = A \sin\{\Omega t + \Phi + \Delta\varphi\lambda(t)\}. \quad (2.31)$$

Пусть модулирующей функцией является синусоида $\lambda(t) = \sin \omega_1 t$, тогда фазомодулированный сигнал будет описываться выражением :

$$s(t) = A[\sin(\Omega t + \Phi)\cos(\Delta\varphi \sin \omega_1 t) + \cos(\Omega t + \Phi)\sin(\Delta\varphi \sin \omega_1 t)]. \quad (2.32)$$

Отсюда видно, что ЧМ и ФМ-сигналы похожи по форме. Различие заключается лишь в том, что коэффициент модуляции для ФМ-сигнала $\Delta\varphi$ постоянен, а индекс модуляции для ЧМ-сигнала β зависит от частоты модулирующего сигнала ω_1.

При использовании ФМ для передачи данных каждому информационному элементу – биту – ставится в соответствие определенное значение фазы (например, 0° – для передачи нуля, 180° – для передачи единицы).

При *фазоразностной модуляции* (DPSK – Differential Phase Shift Keying) каждому информационному элементу ставится в соответствие не абсолютное значение фазы, а ее изменение относительно предыдущего значения. Если информационный элемент есть *дибит*, то в зависимости от его значения (00, 01, 10 или 11) фаза сигнала может измениться на 90, 180, 270° или не измениться вовсе. Из теории информации известно, что фазовая модуляция наиболее информативна, однако увеличение числа кодируемых бит выше трех (8 позиций поворота фазы) приводит к резкому снижению помехоустойчивости. Поэтому в высокоскоростных модемах применяются комбинированные амплитудно-фазовые методы модуляции.

Квадратурно-амплитудная модуляция. *Многопозиционную амплитудно-фазовую модуляцию* называют еще *квадратурной амплитудной модуляцией* (QAM – Quadrature Amplitude Modulation). В данном виде модуляции для повышения пропускной способности используют одновременную манипуляцию двух параметров несущего колебания – амплитуды и фазы. Каждое возможное состояние модулированного сигнала (вектор сигнала или точка сигнального пространства) характеризуется определенным значением амплитуды и фазы, которые входят в так называемое *созвездие*.

В настоящее время используют модуляции, в которых количество кодируемых на одном бодовом интервале информационных бит может доходить до 8, а, соответственно, созвездие иметь число состояний сигнала в сигнальном пространстве – до 256.

Однако с ростом модуляционной скорости и числа состояний сигнала устойчивость к помехам многопозиционной QAM-модуляции быстро снижается, что связано с уменьшением энергии элемента сигнала и различий между соседними допустимыми состояниями сигналов. Значительное повышение реальной помехоустойчивости на скоростях передачи 9600 бит/с и более было достигнуто, благодаря применению комбинации модуляции с решетчатым кодированием.

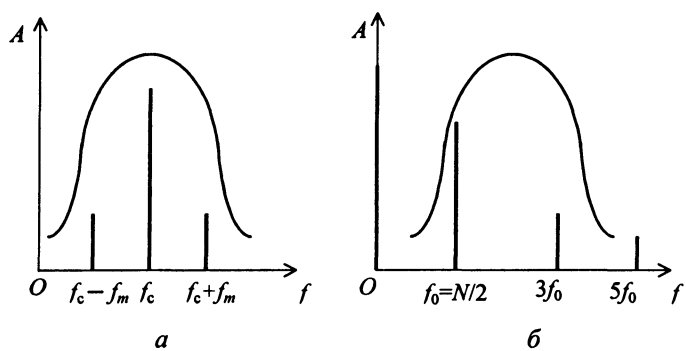
Модуляция с решетчатым кодированием. В современных высокоскоростных протоколах используют так называемую модуляцию с *решетчатым кодированием или треллис-кодированием* (TCM – Trellis Coded Modulation), которая позволяет повысить помехозащищенность передачи информации и снизить требования к отношению сигнал/шум в канале от 3 до 6 дБ. Суть этого кодирования заключается во введении избыточности в пространство сигналов за счет чего создаются корреляционные связи между передаваемыми символами. Пространство сигналов расширяется вдвое путем добавления к информационным битам еще одного, который образуется посредством сверточного кодирования над частью информационных бит и введения элементов запаздывания. Расширенная таким образом группа подвергается все той же многопозиционной амплитудно-фазовой модуляции. В процессе демодуляции принятого сигнала проводится его декодирование по весьма изощренному алгоритму Витерби, позволяющему за счет введенной избыточности и знания предыстории выбрать по критерию максимального правдоподобия из сигнального пространства наиболее достоверную точку и, тем самым, определить значения информационных бит.

Если все принято без ошибок, то треллис-бит просто удаляется. А вот если ошибки были, то с очень большой вероятностью последовательность, содержащая сбойные биты, окажется запрещенной. При помощи специального итеративного алгоритма осуществляется поиск по решетке (отсюда и название). Декодер Витерби находит «наиболее подходящую» разрешенную последовательность и заменяет ею сбойную. Причем, с весьма большой вероятностью эта замена действительно окажется верной. Треллис-коды построены таким образом, что они защищены от перепутывания именно соседних состояний в

пространстве сигналов, которые как раз и рискуют «перепутаться» в результате воздействия помехи.

Амплитудно-фазовая модуляция с несколькими несущими. Один из современных методов амплитудно-фазовой модуляции основан на одновременной передаче множества несущих. Например, в одном конкретном приложении, используют 48 несущих, разделенных полосой в 45 Гц. Путем комбинирования фазовой и амплитудной модуляции, каждая несущая может иметь до 32 дискретных состояний на каждый период бода, позволяя переносить 5 бит на бод. Таким образом, 48 несущих могут переносить 5×48 = 240 бит на бод. Для работы со скоростью 9600 бит/с скорость модуляции требует только 40 бод (9600:240); такая низкая скорость весьма терпима к фазовым и амплитудным скачкам, которые присущи телефонной сети. Реально используемая полоса составляет 2240 Гц. Модуляция и демодуляция осуществляются в цифровом виде в микропроцессоре. Этот метод модулирования иллюстрирует, что достаточно дешевая электроника позволяет реализовывать идеи, которые никогда не стали бы практикой совсем недавно.

Спектр модулированною сигнала. Как было сказано выше, спектр результирующего модулированного сигнала зависит от типа модуляции и скорости модуляции, т. е. желаемой скорости передачи исходной информации. При амплитудной модуляции спектр состоит из синусоиды несущей частоты $f_c = \Omega$ двух боковых гармоник: $(f_c + f_m)$ и $(f_c - f_m)$, где $f_m = \omega$ – частота изменения информативного параметра синусоиды, совпадающая со скоростью передачи данных при использовании двух уровней амплитуды (рис. 2.8, *а*). Частота f_m определяет пропускную способность линии при данном способе кодирования. При небольшой частоте модуляции ширина спектра сигнала будет также небольшой (равной $2f_m$), поэтому сигналы не будут искажаться в канале, если его полоса пропускания будет больше или равна $2f_m$. Для канала тональной частоты такой способ модуляции приемлем при скорости передачи данных не больше 3100/2=1550 бит/с. Если же для представления данных используют четыре уровня амплитуды, то пропускная способность канала повышается до 3100 бит/с.



а *б*

Рис. 2.8. Спектры сигналов при амплитудной модуляции (*а*) и потенциальном кодировании (*б*)

При ФМ и ЧМ спектр сигнала получается более сложным, чем при АМ, так как боковых гармоник здесь образуется более двух, но они также симметрично расположены относительно основной несущей частоты, а их амплитуды быстро убывают. Поэтому эти виды модуляции также хорошо подходят для передачи данных по каналу тональной частоты.

Рассмотрим 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. В данном случае значения бод и бит в 1с совпадают.

Для потенциального кодирования спектр непосредственно получается по формулам 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 $f_0 = N/2$. Амплитуды этих гармоник убывают достаточно медленно – с коэффициентами 1/3, 1/5, 1/7.... от амплитуды гармоники f_0 (рис. 2.8, *б*). 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 0 Гц 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 используют.

Цифровое кодирование. При цифровом кодировании дискретной информации применяют потенциальные и импульсные коды. В потенциальных кодах для представления логических единиц и нулей используют только значение потенциала сигнала, а его перепады не учитывают. Импульсные коды позволяют представить двоичные данные либо импульсами определенной полярности, либо фронтом импульса – перепадом потенциала определенного направления. При использовании прямоугольных импульсов для передачи дискретной информации необходимо выбрать такой способ кодирования, который одновременно:

- имел при одной и той же битовой скорости наименьшую ширину спектра результирующего сигнала. Более узкий спектр сигналов позволяет в канале с

одной и той же полосой пропускания получать более высокую скорость передачи данных. Кроме того, часто к спектру сигнала предъявляется требование отсутствия постоянной составляющей, т. е. наличия постоянного тока между передатчиком и приемником. В частности, применение различных трансформаторных схем *гальванической развязки* препятствует прохождению постоянного тока;

• обеспечивал синхронизацию между передатчиком и приемником. Синхронизация передатчика и приемника нужна для того, чтобы приемник точно знал, в какой момент времени необходимо считывать новую информацию с линии связи. Эта проблема в сетях связи 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. 2.9), в которой 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 позже или раньше соответствующего сигнала данных и бит данных будет пропущен или считан повторно. Поэтому в сетях применяют так называемые *самосинхронизирующиеся коды,* сигналы которых несут для передатчика указания о том, в какой момент времени нужно осуществить распознавание очередного бита (или нескольких бит, если код ориентирован более чем на два состояния сигнала). Любой резкий перепад сигнала – так называемый фронт – служит хорошим указанием для 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 появления входного кода;

• обладал способностью распознавать ошибки и низкой стоимостью реализации.

Распознавание и коррекцию искаженных данных сложно осуществить средствами физического уровня, поэтому чаще всего эту работу берут на себя протоколы верхних уровней: канального, сетевого, транспортного или прикладного. С другой стороны, распознавание ошибок на физическом уровне экономит время, так как приемник не ждет полного помещения кадра в буфер, а отбраковывает его сразу при распознавании ошибочных бит внутри кадра.

Рис. 2.9. Синхронизация приемника и передатчика на небольших расстояниях

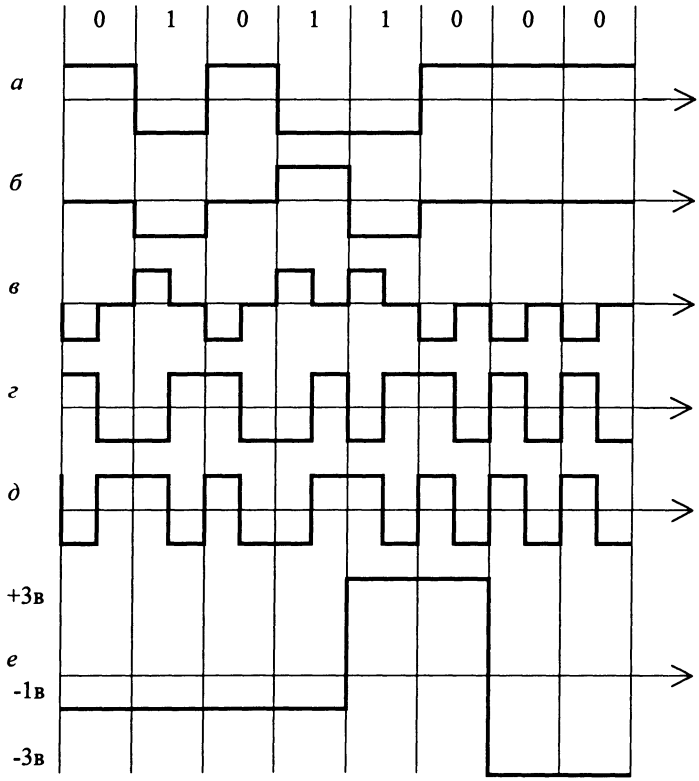


Рис. 2.10. Дискретные формы представления сигналов в канале связи

Требования, предъявляемые к методам кодирования, являются взаимно противоречивыми, поэтому каждый из рассматриваемых ниже популярных методов цифрового кодирования обладает своими преимуществами и недостатками.

На рис. 2.10, *а* показан метод потенциального кодирования, называемый также кодированием *без возвращения к нулю* (NRZ – Non Return to Zero). Название отражает суть метода, в котором при передаче последовательности единиц сигнал не возвращается к нулю в течение такта (как мы увидим ниже, в других методах кодирования возврат к нулю в этом случае происходит). Метод NRZ прост в реализации, хорошо распознает ошибки (из-за двух резко отличающихся потенциалов), но не обладает самосинхронизацией. При передаче длинной последовательности единиц или нулей сигнал в среде передачи не изменяется, поэтому приемник не может определить по входному сигналу моменты времени, когда нужно в очередной раз считывать данные. Даже при наличии высокоточного тактового генератора приемник может ошибиться моментом считывания данных, так как частоты двух генераторов 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.

Другим серьезным недостатком метода NRZ является наличие низкочастотной составляющей, которая приближается к нулю при передаче длинных последовательностей единиц или нулей. Из-за этого многие каналы связи, не обеспечивающие прямого гальванического соединения между приемником и источником, этот вид кодирования не поддерживают. В сетях используют различные модификации кода NRZ, в которых устранены плохая самосинхронизация и постоянная составляющая этого кода. Достоинством кода NRZ, из-за чего имеет смысл его улучшать, является низкая частота основной гармоники f_0, равная $N/2$ Гц (см. § 2.1). В других методах кодирования, например манчестерском, основная гармоника имеет более высокую частоту.

Одной из модификаций метода NRZ является метод *биполярного кодирования с альтернативной инверсией* (AMI – Bipolar Alternate Mark Inversion). В этом методе (рис. 2.10, *б*) используют три уровня потенциала – отрицательный, нулевой и положительный. Логический нуль кодируется нулевым потенциалом, а логическая единица – либо положительным, либо отрицательным потенциалом, при этом потенциал каждой новой единицы противоположен потенциалу предыдущей.

Код AMI частично решает проблему постоянной составляющей и отсутствия самосинхронизации, присущие коду NRZ. Это происходит при передаче длинных последовательностей единиц. При этом сигнал представляет собой последовательность разнополярных импульсов с тем же спектром, что и у кода NRZ, передающего чередующиеся нули и единицы, т. е. без постоянной составляющей и с основной гармоникой $N/2$ Гц (где N – битовая скорость передачи данных). Длинные последовательности нулей также опасны для кода AMI, как и для кода NRZ – сигнал вырождается в постоянный потенциал нулевой амплитуды. Поэтому код AMI требует дальнейшего улучшения, хотя задача упрощается – осталось справиться только с последовательностями нулей.

Для различных комбинаций бит 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 строгого чередования полярности сигналов говорит о ложном импульсе или исчезновении с линии корректного импульса. Сигнал с некорректной полярностью называется *запрещенным сигналом (signal violation)*.

В коде AMI используют три уровня сигнала на линии. Дополнительный третий уровень требует увеличение мощности передатчика примерно на 3 дБ для обеспечения той же достоверности приема бит на линии, что является общим недостатком кодов с несколькими состояниями сигнала, в отличие от кодов, которые различают только два состояния.

Существует код, похожий на AMI, с двумя уровнями сигнала. При передаче нуля такой код передает потенциал, который был установлен в предыдущем такте (т. е. не меняет его), а при передаче единицы потенциал инвертируется на противоположный. Этот код называется *потенциальным кодом с инверсией при единице* (NRZI – Non Return to Zero with ones Inverted). Он удобен в случаях, когда использование третьего уровня сигнала весьма нежелательно, например, в оптических кабелях, где устойчиво распознаются два состояния сигнала – свет и темнота.

Улучшить потенциальные коды, подобные AMI и NRZI, можно двумя методами. Первый метод основан на добавлении в исходный код избыточных бит, 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. Но 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 вероятности появления единиц и нулей становились близкими. 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 бит. Избыточные биты при этом по линии не передаются. Оба метода относятся к логическому кодированию, и форму сигналов на линии они не определяют.

Биполярный импульсный код. В импульсных кодах данные представлены полным импульсом или же его частью – фронтом. Наиболее простым случаем является *биполярный импульсный код,* в котором единица представлена импульсом одной полярности, а ноль – другой (рис. 2.10, *в*). Каждый импульс длится половину такта. Этому коду присущи хорошие самосинхронизирующиие свойства, 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 используют редко.

В локальных сетях до недавнего времени самым распространенным методом кодирования был так называемый *манчестерский код* (рис. 2.10, *г*). Его применяют в технологиях Ethernet и Token Ring. В манчестерском коде для кодирования единиц и нулей используется перепад потенциала, т. е. фронт импульса. При этом 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 – обрат-

ным перепадом. Если нужно представить несколько единиц или нулей подряд, то в начале каждого такта происходит служебный перепад сигнала. Разновидностью манчестерского кода является *дифференциальный манчестерский код* (рис. 2.10, *д*). Здесь наличие фронта в начале такта соответствует передаче нуля, а его отсутствие – передаче единицы. Середина каждого такта отводится для служебных синхронизирующих перепадов сигнала.

Манчестерские коды отличаются хорошими самосинхронизирующими свойствами, так как сигнал изменяется по крайней мере один раз за такт передачи одного бита данных. Полоса пропускания манчестерского кода уже, чем у биполярного импульсного. У него отсутствует постоянная составляющая, 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 или NRZ.

Потенциальный код 2B1Q (рис. 2.10, *е*). Название этого кода отражает его суть – каждые два бита (2B) передаются за один такт сигналом, имеющим четыре состояния (1Q). Паре бит 00 соответствует потенциал –2,5В, паре бит 01 соответствует потенциал –0,833В, паре 11 – потенциал +0,833В, а паре 10 – потенциал +2,5В. При кодировании потенциальным кодом необходимы дополнительные меры по борьбе с длинными последовательностями одинаковых пар бит, так как при этом сигнал превращается в постоянную составляющую потенциального кода. При случайном чередовании бит спектр сигнала в два раза уже, чем у кода NRZ, так как при той же битовой скорости длительность такта увеличивается в два раза. 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 сигнала четко различались приемником на фоне помех.

Дискретная модуляция аналоговых сигналов. Одной из основных тенденций развития сетевых технологий является передача в одной сети как дискретных, так и аналоговых по своей природе сигналов. Источниками дискретных сигналов являются компьютеры и вычислительные устройства, а источниками аналоговых сигналов – телефоны, видеокамеры, звуко- и видеовоспроизводящая аппаратура.

На ранних этапах решения этой проблемы в территориальных сетях все типы данных передавались сигналами в аналоговой форме, при этом дискретные по своему характеру компьютерные данные преобразовывались в аналоговую форму при помощи модемов. Однако по мере развития техники съема и передачи аналоговых данных выяснилось, что передача их в аналоговой форме не позволяет улучшить качество принятых на другом конце линии данных, если они существенно исказились при передаче. Сам аналоговый сигнал не дает никаких указаний ни о том, что произошло искажение, ни о том, как его исправить, поскольку форма аналогового сигнала может быть любой, в том числе и такой, которую зафиксировал приемник. Улучшение же качества линий, особенно территориальных, требует огромных усилий и капиталовложений.

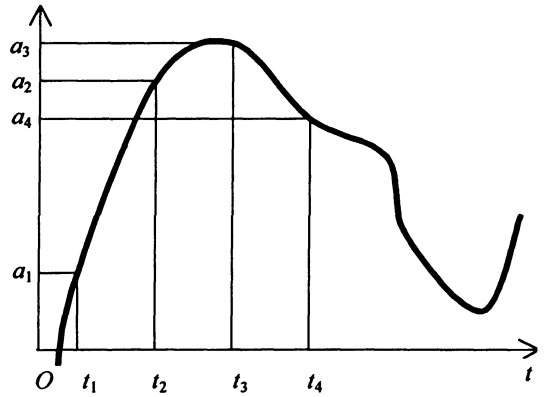


Рис. 2.11. Диаграмма ИКМ:
$a_1, a_2, ..., a_4$ – амплитуды оцифрованного сигнала, $_1, t_2, ...$ – моменты квантования ($t_2 - t_1 = t_4 - t_3 = T$ – шаг квантования)

Поэтому на смену аналоговой технике записи и передачи звука и изображения пришла цифровая техника, которая использует так называемую дискретную модуляцию исходных непрерывных во времени аналоговых процессов.

Дискретные способы модуляции основаны на дискретизации непрерывных процессов как по амплитуде, так и по времени (рис. 2.11). Рассмотрим принципы дискретной модуляции на примере *импульсно-кодовой модуляции* ИКМ (PCM – Pulse Code Modulation), которая широко применяется в цифровой телефонии.

Амплитуда исходной непрерывной функции измеряется с заданным периодом, что обеспечивает дискретизацию по времени. Затем каждый замер представляется в виде двоичного числа определенной разрядности, что означает дискретизацию по значениям функции – 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 каналам связи в виде последовательности единиц и нулей. Для этого применяют те же методы кодирования, что и в случае передачи изначально дискретной информации, например, методы, основанные на коде 2B1Q.

На приемной стороне линии коды преобразуются в исходную последовательность бит в *цифро-аналоговом преобразователе* (ЦАП), который демодулирует оцифрованные амплитуды непрерывного сигнала, восстанавливая исходную непрерывную функцию времени.

Дискретная модуляции основана на *теории отображения Найквиста–Котельникова.* 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 существенно отличается от исходной.

Импульсно-кодовая модуляция состоит из двух действий.

1. Берутся отсчеты амплитуды входного сигнала через какие-то интервалы времени (обычно равные, см. рис. 2.11). По *теореме Найквиста–Котельникова* эти интервалы должны удовлетворять следующему неравенству:

$$T \leq 1/2F, \tag{2.33}$$

где F – частота наивысшей гармоники; T – интервал или шаг квантования.

Соответственно, чем меньше интервал T, тем выше качество преобразования.

2. Полученные отсчеты амплитуды квантуются по уровню и кодируются. Квантование представляет собой округление мгновенного значения амплитуды, т. е. замену этого значения одним из разрешенных значений, отстоящих друг от друга на конечные интервалы. Шкала разрешенных значений называется шкалой квантования, а интервал между значениями – шагом квантования (см. рис. 2.11). Чем меньше шаг квантования, тем выше качество преобразования. Квантование осуществляет АЦП и его быстродействием определяется минимальный возможный шаг квантования. Шкала квантования, в свою очередь, определяется разрядностью АЦП. Например, 8-разрядный преобразователь может квантовать амплитуду на 256, а 16-разрядный – на 65536 интервалов. Таким образом, в заданном входном диапазоне 8-разрядный АЦП заметит отклонение аналогового сигнала, если тот изменится не менее, чем на 1/256 часть своего максимального значения. Следовательно, по линии передаются не сами значения амплитуды сигнала, а номера уровней, что и составляет сущность ИКМ.

Кодирование предполагает замену значения квантованного сигнала 8-разрядным словом. Такая последовательность операций обеспечивает представление сериями 8-разрядных слов (кодов) любого изменяемого во времени аналогового сигнала.

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 для компьютерных данных (более подробно рассмотрены ниже): вычисление контрольной суммы, повторная передача искаженных кадров, применение самокорректирующихся кодов. Для качественной передачи голоса в ИКМ используют частоту квантования амплитуды звуковых колебаний в 8000 Гц. Это связано с тем, что в аналоговой телефонии для передачи голоса был выбран диапазон от 300 до 3400 Гц, который достаточно качественно передает все основные гармоники голосов. В соответствии с теоремой Найквиста–Котельникова для качественной передачи голоса достаточ-

но выбрать частоту дискретизации, в два раза превышающую самую высокую гармонику непрерывного сигнала, т. е. 2 × 3400 = 6800 Гц. Выбранная в действительности частота дискретизации 8000 Гц обеспечивает некоторый запас качества. В ИКМ обычно используют 7 или 8 бит кода для представления амплитуды одного замера. Соответственно это дает 127 или 256 градаций звукового сигнала, что оказывается вполне достаточным для качественной передачи голоса.

При использовании ИКМ для передачи одного голосового канала необходима пропускная способность 56 или 64 кбит/с в зависимости от того, каким количеством бит представляется каждый замер. Если для этого используют 7 бит, то при частоте передачи замеров в 8000 Гц получаем:

$$8000 \times 7 = 56000 \text{ бит/с или } 56 \text{ кбит/с};$$

в случае 8 бит:

$$8000 \times 8 = 64000 \text{ бит/с или } 64 \text{ кбит/с}.$$

Стандартным является цифровой канал 64 кбит/с, называемый *элементарным каналом цифровых телефонных сетей.*

Передача непрерывного сигнала в дискретном коде требует от сетей жесткого соблюдения временного интервала в 125 мкс (соответствующего частоте дискретизации 8000 Гц) между соседними замерами, т. е. синхронной передачи данных между узлами сети. При несоблюдении 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 потере распознаваемости произносимых слов.

В то же время потеря одного замера при соблюдении синхронности между остальными замерами практически не сказывается на воспроизводимом звуке после цифро-аналогового преобразования на приемной стороне. Это происходит за счет сглаживающих устройств в цифро-аналоговых преобразователях (ЦАП), основанных на свойстве инерционности любого физического процесса, – амплитуда звуковых колебаний не может мгновенно измениться на большую величину.

На качество сигнала на выходе ЦАП влияет не только синхронность поступления на его вход замеров, но и погрешность дискретизации амплитуд этих замеров. Воспроизводимый в приемнике сигнал не совпадает в точности с оригинальным сигналом. Дело в том, что из-за конечного числа уровней квантования (256) вершина дискрета может занимать произвольное положение внутри интервала, который определяется величиной шага квантования, т. е. расстоянием между последовательными уровнями квантования. В приемнике же значение восстановленного сигнала располагается в середине интервала квантования. Разность между оригинальным сигналом и восстановленным на приемной стороне называется *шумом квантования.*

Существуют и другие методы дискретной модуляции, позволяющие представить замеры голоса в более компактной форме, например, в виде последовательности 4- или 2-битных чисел. При этом один голосовой канал требует меньшей пропускной способности, например 32, 16 кбит/с или меньше. С 1985 г. применяется стандарт CCITT кодирования голоса, (ADPCM – Adaptive Differential Pulse Code Modulation). Коды ADPCM основаны на нахождении разностей между последовательными замерами голоса, которые затем и передаются по сети. В этом коде для хранения одной разности используют 4 бит и голос передается со скоростью 32 кбит/с. Метод Linear Predictive Coding (LPC) делает замеры исходной функции более редко и базируется на прогнозировании направления изменения амплитуды сигнала. При помощи этого метода можно понизить скорость передачи голоса до 9600 бит/с.

Представленные в цифровой форме непрерывные сигналы легко можно передать через компьютерную сеть. Для этого достаточно поместить несколько замеров в кадр какой-нибудь стандартной сетевой технологии, снабдить кадр правильным адресом назначения и отправить адресату. Адресат должен извлечь из кадра замеры и подать их с частотой квантования (для голоса – с частотой 8000 Гц) на ЦАП. По мере поступления следующих кадров с замерами голоса операция должна повторяться. Если кадры будут прибывать синхронно, то качество голоса будет высоким. Однако кадры в компьютерных сетях могут задерживаться как в конечных узлах (при ожидании доступа к разделяемой среде), так и в промежуточных коммуникационных устройствах: мостах, коммутаторах и маршрутизаторах. Поэтому качество голоса при передаче в цифровой форме через компьютерные сети обычно бывает невысоким. Для качественной передачи оцифрованных непрерывных сигналов (голоса, изображения) сегодня используют специальные цифровые сети, такие как ISDN, ATM, и сети цифрового телевидения. Для передачи внутрикорпоративных телефонных разговоров в настоящее время популярны сети Frame relay, задержки передачи кадров которых укладываются в допустимые пределы.

2.2. Методы защиты от ошибок и сжатия данных

Принципы помехоустойчивого кодирования

Под кодированием информации будем понимать преобразование формы представления информации с целью обеспечения удобства ее передачи по каналам связи или хранения. Правило, по которому осуществляется кодирование, называется *кодирующим отображением* или *кодом*. Пусть $A = \{p, q, r, s\}$ является входным алфавитом кода Γ, а $B = \{a, b\}$ – его выходным алфавитом. Код Γ в процессе кодирования перерабатывает слово над алфавитом A в слово над алфавитом

Таблица 2.1. **Пример задания кода**

p	*aa*
q	*ab*
r	*ba*
s	*bb*

B. Если код *Г* описывается табл. 2.1, то слово *prq* в алфавите *A* преобразуется в алфавите *B* в слово *aabaab*. Слова, сопоставляемые элементам множества *A* по правилу *Г* в алфавите *B*, называются *кодовыми комбинациями*. Если $x \in A$ и $Гx = a_1a_2...a_n$, где $a_i \in B$ для всех *i*, то говорят что символу *x* соответствует кодовая комбинация $a_1a_2...a_n$ (иногда эту кодовую комбинацию называют *кодом символа*).

Коды, формирующие кодовые комбинации различной длины, называются *неравномерными*, а коды, которым соответствуют кодовые комбинации равной длины – *равномерными*.

Значность кода – длина кодовых комбинаций равномерного кода.

Декодирование – это процесс обратный кодированию, т. е. замена кодовой комбинации символом из входного алфавита. Если процесс кодирования осуществляется по правилу *Г*, то процесс декодирования основан на правиле $Г^{-1}$ – отображении, обратном *Г*. Пусть α – слово в алфавите *A*, $\beta = Г\alpha$ – слово в алфавите *B*. Код называется обратимым, если для любого слова $\beta = Г\alpha$ в алфавите *B* существует единственное преобразование $Г^{-1}\beta = \alpha$, т. е. по слову β в алфавите *B*, являющимся последовательностью кодовых комбинаций, кодирующих слово α, всегда однозначно восстанавливается слово α. Для того, чтобы код был обратимым, необходимо выполнение двух *условий обратимости кода*:

- разным символам входного алфавита *A* должны быть сопоставлены различные кодовые комбинации;
- никакая кодовая комбинация не должна составлять начальной части какой-либо другой кодовой комбинации.

Выполнение второго условия необходимо только для неравномерных кодов, для равномерных кодов оно выполняется автоматически при выполнении первого условия.

В системах передачи дискретных сообщений (данных) используют два алфавита: один имеет достаточно большой объем, применяется для представления сообщения на языке источника и получателя информации и называется *внешним алфавитом*; второй используют непосредственно для передачи информации по каналу, он содержит небольшое число символов и называется *внутренним алфавитом*. Чем меньше символов содержит внутренний алфавит, тем легче их различать в условиях помех. Проблемы помехоустойчивого кодирования решаются специальными методами.

Рассмотрим представление кодовых комбинаций применительно к равномерным кодам, в частности к *блочным*, в которых кодовые комбинации кодируются и декодируются независимо друг от друга. Пусть выходной алфавит *B* равномерного *n*-значного кода *Г* состоит из *m* символов; число *m* называется основанием кода. Кодовая комбинация такого кода имеет вид $\alpha_1, \alpha_2, ..., \alpha_n$, где α_i – значение *i*-го разряда кода, $i = 1, 2, ..., n$; $\alpha_i \in B$.

Упорядочим (произвольно, но раз и навсегда) символы алфавита $B = (C_0, C_1, ..., C_{m-1})$ и будем под ними понимать различные классы вычетов по модулю m. Индекс класса при этом поставим в соответствие значению остатка при делении любого представителя класса на число m.

Введем на множестве B две алгебраические операции: умножение и сложение, понимая под произведением классов $C_k C_j$ класс C_r (где r – остаток от деления произведения kj на m) и под суммой классов $C_k + C_j$ класс C_{k+j} при $k + j < m$ и класс C_{k+j-m} при $k + j \geq m$.

Кодовые комбинации над выходным алфавитом B n-значного равномерного кода можно рассматривать как векторы в n-мерном векторном линейном пространстве над полем B. Это пространство в дальнейшем будем называть кодовым пространством, а его элементы – кодовыми векторами. Для упрощения оперирования с классами вычетов по модулю m в дальнейшем будем обозначать их наименьшими представителями 0, 1, 2, ..., $m - 1$.

При анализе воздействия ошибок на кодовые векторы в кодовом пространстве вводится метрика. Наибольший практический интерес представляет метрика Хэмминга. Вес вектора v по Хэммингу равен числу ненулевых разрядов этого вектора, а *расстояние Хэмминга* между векторами v_j и v_k определяется как вес разности векторов v_j и v_k. Для бинарных кодов ($m = 2$) имеем

$$d(v_j, v_k) = \sum_{i=1}^{n} (a_i \oplus b_i). \qquad (2.34)$$

Кодовое пространство n-разрядного кода с основанием m составляет m^n векторов. При передаче информации, как правило, используются не все возможные комбинации, а лишь некоторое их подмножество $V_1 = \{v_1, ..., v_N\}$, где $N < m^n$. Обозначим расстояние между парой векторов набора V_1 символом $d(v_i, v_j)$, $i, j = 1, 2, ..., N$, $i \neq j$. Величина $\min d(v_i, v_j)$, представляющая собой минимальное расстояние между парой векторов набора V_1, называется *кодовым расстоянием* и обозначается символом d.

В системах передачи информации в основном используют бинарные коды, т. е. коды с основанием $m = 2$: B = {0, 1}. Учитывая это, главное внимание в дальнейшем будем уделять бинарным кодам.

Модели ошибок. В моделях систем передачи информации используется дискретный канал связи (рис. 2.12). Передаваемый кодовый вектор v_1 складывается в дискретном канале поразрядно по модулю 2 с вектором ошибки e, и в общем на выходе образуется уже другой, искаженный кодовый вектор v_1'. Например, если $v_1 = 11111$ и $e = 01000$, то $v_1' = v_1 \oplus e = 11111 + 01000 = 10111$, т. е. во втором разряде результирующей кодовой комбинации v_1', как видно, произошла ошибка. Таким образом, в разрядах передаваемой кодовой комбинации, соответствующих единичным разрядам вектора e, возникают ошибки.

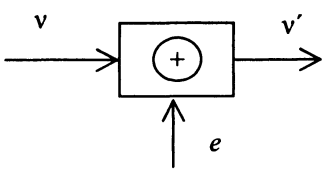


Рис. 2.12. Схема дискретного канала

При теоретических исследованиях процесса возникновения ошибок в дискретном канале используют математические модели ошибок. Под математической моделью ошибки понимается распределение вероятностей по всем возможным векторам ошибки. В соответствии с принятыми моделями ошибок различают и дискретные каналы (ДК). Дискретный канал называется *стационарным или однородным* дискретным каналом без памяти, если условные вероятности того, что на j-й позиции кодовой комбинации принят символ y_j, при условии, что на i-й позиции на вход канала подан символ x_i, для всех позиций j одинаковы и не зависят от времени и от значений x_i и y_j на других позициях кодовой комбинации: $p(y_j|x_i) = p(y_r|x_k)$.

В качестве примера рассмотрим одну из наиболее часто встречающихся моделей ошибки, которая основана на следующей статистической гипотезе: в каждом разряде вектора ошибки единица появляется с вероятностью p независимо от того, какие значения получили остальные разряды вектора ошибки. Такой стационарный ДК, в котором вероятности искажения любого символа кодовой комбинации одинаковы, называется *симметричным* каналом без памяти.

Назовем величину, равную числу единиц в векторе ошибки, кратностью ошибки и обозначим символом q. В теории вероятностей доказывается, что выдвинутой статистической гипотезе отвечает биномиальный закон распределения кратности ошибки. Таким образом, для рассматриваемого примера математической моделью ошибки может служить выражение

$$P_{n,q} = C_n^q p^q (1-p)^{n-q}.$$

Здесь $P_{n,q}$ – вероятность того, что при передаче по дискретному каналу в кодовой комбинации бинарного кода длины n появится ошибка кратности q.

Значительно больший практический интерес представляют симметричные каналы с памятью, в которых условные вероятности $p(y_j|x_i)$ для каждой пары i j зависят как от времени, так и от переходов, имевших место ранее.

Подавляющее число реальных каналов связи имеет склонность к многоступенчатому группированию ошибок, в чем и выражается запоминание некоторого состояния канала. При описании группирования ошибок с помощью простой цепи Маркова канал представляется набором состояний s_i, которые переходят друг в друга с вероятностью p_{ij} и в каждом из которых ошибки независимы и происходят с вероятностью P_i. Простейшей моделью такого типа ошибки является модель Гильберта (рис. 2.13).

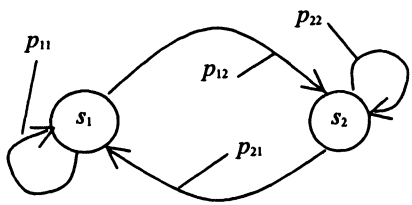


Рис. 2.13. Модель Гильберта

В состоянии s_1 ошибки отсутствуют $P_1 = 0$, а в состоянии s_2 ошибки появляются с вероятностью $P_2 \neq 0$. Если известны вероятности

перехода $p_{11}, p_{12}, p_{21}, p_{22}$, то статистика ошибок образует простую марковскую цепь последовательности состояний с матрицей переходов:

$$P(s) = \begin{Vmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{Vmatrix}.$$

Чтобы выполнялось условие группирования ошибок в канале, переходные вероятности состояний должны быть значительно меньше вероятностей сохранения состояний, т. е. $p_{12} << p_{11}$; $p_{21} << p_{22}$. Тогда вероятности пребывания канала в состояниях s_1 и s_2 соответственно будут равны:

$$p_1 \cong \frac{p_{21}}{p_{12}+p_{21}}; \; p_2 \cong \frac{p_{12}}{p_{12}+p_{21}},$$

а вероятность ошибки символа:

$$p = P_2 p_2 = P_2 \frac{p_{12}}{p_{12}+p_{21}}. \qquad (2.35)$$

Математические модели ошибок должны отражать реальные процессы, происходящие в канале связи, и строиться на статистике помех. Чем точнее математическая модель описывает действительность, тем точнее можно получить оценки относительно спроектированного кода.

Необходимо учитывать, что эффективность того или иного помехоустойчивого кода всегда зависит от вида помех, действующих в канале связи. Код может быть весьма эффективным (в том смысле, что число необнаруженных ошибок при его применении будет очень мало) при одной статистике помех и очень плохим – при другой. Поэтому при проектировании помехоустойчивых кодов необходимо ориентироваться на определенный вид помех и в соответствии с этим в качестве исходной иметь определенную модель ошибок.

Обнаружение ошибок. Наибольшее распространение при передаче дискретных сообщений получили блочные равномерные коды. Рассмотрим на примере этих кодов как обнаруживаются ошибки. Помехоустойчивость блочных кодов, как и других кодов, достигается введением избыточности в кодовые комбинации. Коды, не обладающие избыточностью, не способны обнаруживать и тем более исправлять ошибки.

В безызбыточных равномерных кодах длины k все 2^k возможных кодовых комбинаций используются, т. е. любой из 2^k кодовой комбинации сопоставляется какой-либо символ внешнего алфавита. Такие коды получили название первичных кодов. Ошибка любой кратности в какой-либо кодовой комбинации всегда приводит к ошибочному декодированию этой кодовой комбинации. Нетрудно видеть, что кодовое расстояние для первичного кода равно единице, т. е. некоторые пары кодовых комбинаций первичного кода располагаются на минимальном расстоянии, отличном от нуля. Для обеспечения помехоустойчивости кода вводят дополнительные разряды. Если, например, для кодирования всех символов внешнего алфавита достаточно иметь k-разрядный первичный код, то

для обеспечения помехоустойчивости к разрядам первичного кода добавляется r избыточных разрядов. При этом длина результирующей кодовой комбинации становится равной $n = k + r$.

Различают избыточные блочные коды разделимые и неразделимые. В *разделимых* кодах роль разрядов кодовых комбинаций разграничена: часть разрядов, часто совпадающая с разрядами исходного первичного кода, являются информационными, остальные разряды играют роль проверочных разрядов. В *неразделимых* кодах все разряды равноправные, и в кодовой комбинации нельзя отделить информационные разряды от проверочных.

В качестве примера неразделимого кода может служить код с постоянным весом «3 из 7». Особенностью этого кода является то, что в любой его кодовой комбинации длины 7 имеется ровно три единицы. Таким образом, всего кодовых комбинаций кода «3 из 7» будет

$$C_7^3 = \frac{7!}{(3! \cdot 4!)} = 35.$$

Обнаруживающая способность данного кода основывается на том, что любая одиночная ошибка изменяет число единиц в кодовой комбинации.

Таким образом, обнаружение ошибок помехоустойчивым кодом возможно благодаря тому, что для передачи информации используются не все 2^n n-разрядные кодовые комбинации равномерного кода, а лишь часть из них. Для разделимых кодов эта часть составляет 2^k кодовых комбинаций, получивших название *разрешенных* кодовых комбинаций. Оставшаяся часть $2^n - 2^k$ кодовых комбинаций, составляющая *запрещенные* кодовые комбинации, при передаче информации не применяется. Использование при кодировании символов внешнего алфавита лишь части кодовых комбинаций позволяет разнести разрешенные кодовые комбинации в кодовом пространстве на расстояние, превышающее единицу. Нетрудно видеть, что если расстояние $d > 1$, то все одиночные ошибки будут переводить разрешенные кодовые комбинации в запрещенные, а появление запрещенной кодовой комбинации на приемной стороне может служить индикатором того, что произошла ошибка.

При разработке реальных кодов учитывают статистику ошибок и требование верности передачи информации. *Верность передачи* оценивается часто как среднее число верно принятых кодовых комбинаций, приходящихся на одну ошибочно принятую кодовую комбинацию, или как вероятность верного приема кодовой комбинации. Так, при выполнении статистической гипотезы о том, что ошибки меньшей кратности появляются чаще ошибок большей кратности, исходя из требования верности передачи, определяют максимальную кратность ошибки, начиная с которой все ошибки меньшей кратности должен обнаруживать помехоустойчивый код. По максимальной кратности ошибки q_m выбирают такое минимальное кодовое расстояние, при котором все разрешенные кодовые комбинации при действии на них ошибок кратностью, не превышающей q_m, переходят в подмножество запрещенных кодовых комбинаций и, следовательно, могут быть обнаружены на приемной стороне системы передачи данных.

Результатом действия ошибки кратности q на разрешенную кодовую комбинацию является новая кодовая комбинация, удаленная от первоначальной на расстояние q. Отсюда следует, что если кодовое расстояние $d \le q$, то при действии ошибки кратности q на какую-либо разрешенную кодовую комбинацию последняя может перейти в другую, но тоже разрешенную кодовую комбинацию и такая ошибка уже не может быть обнаружена. Поэтому для обнаружения всех ошибок, кратность которых не превышает q, кодовое расстояние должно быть больше q: $d > q$. Для обнаружения всех ошибок кратности, не превышающей q_m, кодовое расстояние должно, по крайней мере, на единицу превышать максимальную кратность ошибки: $d = q_m + 1$.

Примером блочного разделимого кода служит код с проверкой на четность. Кодовая комбинация такого кода имеет вид $a_1 a_2 ... a_k b$. Первые k разрядов являются информационными и, как правило, совпадают с разрядами исходного первичного кода. Последний разряд является избыточным и определяется по формуле $b = a_1 \oplus a_2 \oplus ... \oplus a_k$. Из формулы видно, что значение избыточного разряда зависит от того, четное или нечетное число единиц в кодовой комбинации: если число единиц четное, то $b = 0$, в противном случае $b = 1$.

Если выбрать любую кодовую комбинацию первичного кода $a_1 a_2 ... a_k$ и любую другую ближайшую к ней кодовую комбинацию $a'_1 a'_2 ... a'_k$, то, как легко установить, отличие между ними будет лишь в одном разряде, а отсюда следует, что кодовые комбинации будут различной четности. При дополнении этих комбинаций проверочными разрядами последние не будут совпадать, т. е. $b \ne b'$. Следовательно, кодовые комбинации $a_1 a_2 ... a_k b$ и $a'_1 a'_2 ... a'_k b'$ после дополнения разрядами b и b' будут отличаться уже в двух разрядах. Так как данный вывод справедлив для любых двух ближайших кодовых комбинаций исходного первичного кода, то после введения дополнительных разрядов вновь образованный код с проверкой на четность будет иметь кодовое расстояние $d = 2$ и обладать способностью обнаруживать все одиночные ошибки.

Исправление ошибок. Помехоустойчивые коды, позволяющие не только обнаруживать ошибки, но и исправлять их, называются *корректирующими кодами*. Общая идея исправления ошибок кратности не более q_m заключается в следующем. Число возможных кодовых комбинаций M помехоустойчивого кода разбивается на N классов по числу N разрешенных кодовых комбинаций. Разбиение осуществляется таким образом, чтобы в каждый класс входили одна разрешенная кодовая комбинация и ближайшие к ней запрещенные. При декодировании определяется, какому классу принадлежит принятая кодовая комбинация. Если кодовая комбинация принята с ошибкой, т. е. является запрещенной, то она исправляется на разрешенную кодовую комбинацию, принадлежащую тому же классу.

В теории кодирования доказывается, что для обеспечения возможности исправления ошибок кратности не более q_m кодовое расстояние должно быть больше $2q_m$. Обычно оно выбирается по формуле $d = 2q_m + 1$.

Актуальной является задача определения наибольшего числа N разрешенных кодовых комбинаций n-разрядного двоичного кода с кодовым расстоянием d. В теории кодирования существуют следующие отношения:

$d=1$	$N=2^n$
$d=2$	$N=2^{n-1}$
$d=3$	$N \le 2^n(1+n)^{-1}$
$d=2q+1$	$N \le 2^n \cdot \left(1+\sum_{i=1}^{q} C_n^i\right)^{-1}$

Матричное представление [n, k]-кодов. Среди блочных кодов широкое распространение получили линейные коды. Линейными m-ичными кодами называются k-мерные подпространства n-мерного линейного векторного пространства. При этом число n имеет смысл длины кодовой комбинации, число k определяет число информационных разрядов. Линейные коды называют также [n, k]-кодами.

Среди линейных кодов особую роль играют групповые коды, для которых $m = 2$ (двоичные коды). Существуют различные способы задания групповых кодов. Наиболее распространенными являются матричное описание кодов и задание их с помощью порождающих многочленов.

Запишим кодовую комбинацию (кодовый вектор) группового кода длиной n в следующем виде $a_1a_2...a_kb_1b_2...b_r$. Первые k разрядов являются информационными, остальные $r = n - k$ – проверочными. Проверочные символы кодовых комбинаций формируются из информационных символов на основе выражения

$$b_j = c_{j1}a_1 + c_{j1}a_2 + ... + c_{jk}a_k;\ \ j = 1, 2, ..., r. \qquad (2.36)$$

Здесь коэффициенты $c_{j1}, c_{j2}, \ldots, c_{jk}$ принимают значения из множества $\{0,1\}$.

Любая кодовая комбинация, состоящая из k информационных разрядов, все проверочные разряды которой составлены в соответствии с формулой (2.36), является разрешенной кодовой комбинацией [n, k]-кода.

Пусть u и v – две разрешенные кодовые комбинации группового [n, k]-кода. Тогда кодовая комбинация $w = u + v$ также является разрешенной кодовой комбинацией этого кода. Действительно, если

$$u = a_1a_2...a_kb_1b_2...b_r,\ \ v = a_1'a_2'...a_k'b_1'b_2'...b_r', \qquad (2.37)$$

то

$$w = (a_1+a_1')(a_2+a_2')...(a_k+a_k')(b_1+b_1')...(b_r+b_r') = a_1''a_2''...a_k''b_1''b_2''...b_r'', \qquad (2.38)$$

где

$$a_i'' = a_i + a_i',\ \ i = 1, 2, ..., k, \qquad (2.39)$$

$$b_j'' = b_j + b_j' = c_{j1}(a_1 + a_1') + ... + c_{jk}(a_k + a_k'),\ \ k = 1, 2, ..., r. \qquad (2.40)$$

Таким образом, проверочные разряды b_j'' кодовой комбинации w формируются в соответствии с выражением (2.40) и, следовательно, кодовая комбинация w является также разрешенной.

Любые k линейно независимых векторов n-мерного линейного векторного пространства порождают k-мерное подпространство, образуя базис этого подпространства. Отсюда следует, что для задания $[n, k]$-кода достаточно выбрать k любых линейно независимых разрешенных кодовых комбинаций, а остальные разрешенные кодовые комбинации получать как линейные комбинации выбранных базисных векторов. Обычно для задания $[n, k]$-кода используют эту возможность, представляя k линейно-независимых кодовых комбинаций в форме матрицы. Такая матрица называется *порождающей матрицей* $[n, k]$-кода. В общем виде ее можно представить следующим образом:

$$G_{n,k} = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1k} & b_{11} & b_{12} & \dots & b_{1r} \\ a_{21} & a_{22} & \dots & a_{2k} & b_{21} & b_{22} & \dots & b_{2r} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{k1} & a_{k2} & \dots & a_{kk} & b_{k1} & b_{k2} & \dots & b_{kr} \end{vmatrix}. \qquad (2.41)$$

Очевидно, что порождающая матрица $G_{n,k}$ двоичного кода порождает ровно 2^k разрешенных кодовых комбинаций.

В зависимости от выбранного базиса k-мерного подпространства n-мерного кодового пространства кодовое расстояние совокупности 2^k векторов k-мерного подпространства может быть различным. При проектировании $[n,k]$ -кода ставится задача оптимального размещения кодовых векторов в n-мерном кодовом пространстве в соответствии с заданной статистикой ошибок и, в частности, обеспечения максимально возможного кодового расстояния.

Пусть $v_1, v_2, \dots, v_k$ – кодовые векторы-строки, составляющие порождающую матрицу

$$G_{n,k} = \begin{Vmatrix} v_1 \\ v_2 \\ \dots \\ v_k \end{Vmatrix}. \qquad (2.42)$$

Тогда разрешенную кодовую комбинацию $[n, k]$-кода можно представить в виде линейной комбинации векторов:

$$v = g_1 v_1 + g_2 v_2 + \dots + g_k v_k, \qquad (2.43)$$

где $g_1, g_2, \dots, g_k$ – коэффициенты, принимающие значения из множества $\{0,1\}$.

Проверочные разряды $b_1, \dots, b_r$ кодового вектора $v = a_1 a_2 \dots a_k b_1 b_2 \dots b_r$, передаваемого по каналу связи, формируются в соответствии с правилом (2.36).

Это же правило можно использовать на приемном конце канала для проверки правильности кодовой комбинации: равенство (2.36) должно выполняться, если ошибки не произошло. Таким образом, с каждой принятой кодовой комбинацией можно связать систему проверок по числу проверочных разрядов, которая для кодовой комбинации $v = a_1...a_k b_1...b_r$ описывается следующей системой уравнений:

$$\begin{cases} c_{11}a_1 + c_{12}a_2 + ... + c_{1k}a_k + b_1 = 0; \\ c_{21}a_1 + c_{22}a_2 + ... + c_{2k}a_k + b_2 = 0; \\ ..... \\ c_{r1}a_1 + c_{r2}a_2 + ... + c_{rk}a_k + b_r = 0. \end{cases} \quad (2.44)$$

Здесь $c_{ij} \in \{0, 1\}$, $i = 1, ..., r$, $j = 1, ..., k$. Нули в правых частях равенств истолковываются как отсутствие ошибки в принятой кодовой комбинации v.

Для удобства систему проверок (2.36) обычно представляют в матричной форме, а именно как произведение матрицы-строки $v = \| a_1...a_k b_1...b_r \|$, соответствующей принятой кодовой комбинации, на матрицу проверочных коэффициентов:

$$H_{n,k} = \begin{vmatrix} c_{11} & c_{12} & ... & c_{1k} & 1 & 0 & 0 & 0 \\ c_{21} & c_{22} & ... & c_{2k} & 0 & 1 & 0 & 0 \\ ... & ... & ... & ... & ... & ... & ... & ... \\ c_{r1} & c_{r2} & ... & c_{rk} & 0 & 0 & 0 & 1 \end{vmatrix}. \quad (2.45)$$

Матрицу $H_{n,k}$, с помощью которой осуществляется система проверок над принятой кодовой комбинацией, принято называть *проверочной матрицей.*

Система проверок (2.36) над принятой кодовой комбинацией эквивалентна ее умножению на транспонированную проверочную матрицу $H_{n,k}$. Если ошибки нет, то должно выполняться равенство

$$v \times H^{\mathrm{T}}_{n,k} = 0. \quad (2.46)$$

В общем случае результат умножения может быть отличен от нуля:

$$\| a_1...a_k b_1...b_r \| \times H^{\mathrm{T}}_{n,k} = \| c_1 c_2 ... c_i ... c_r \|, \quad (2.47)$$

где $c_i \in \{0, 1\}$, $i = 1, ..., r$.

Матрица-строка $\| c_1...c_r \|$, полученная в результате умножения, называется *синдромом ошибки.* Всего может быть $(2^r - 1)$ различных ненулевых синдромов, разбивающих множество возможных ошибок на $(2^r - 1)$ класса. Это позволяет по виду синдрома ошибки определять, к какому классу относится ошибка. Часто $[n, k]$-код проектируется таким образом, что с вероятностью, близкой к единице, каждый из выделенных $(2^r - 1)$ классов ошибок содержит всего по одному элементу. Такие коды позволяют исправлять ошибки.

Рассматривая разрешенную комбинацию [n, k]-кода как линейную комбинацию кодовых вектор-строк порождающей матрицы и подставляя выражение (2.43) в равенство (2.46), получаем

$$(g_1v_1 + g_2v_2 + ... + g_kv_k,) \times H^{T}_{n,k} = g_1(v_1 \times H^{T}_{n,k}) + \\ + g_2(v_2 \times H^{T}_{n,k}) + ... + g_k(v_k, v \times H^{T}_{n,k}) = 0. \quad (2.48)$$

Из этого выражения очевидно: чтобы для любой разрешенной кодовой комбинации [n, k]-кода выполнялось равенство (2.46), необходимо и достаточно, чтобы выполнялось равенство:

$$\begin{Vmatrix} v_1 \\ v_2 \\ ... \\ v_k \end{Vmatrix} \times H^{T}_{n,k} = 0, \text{ или } G_{n,k} \times H^{T}_{n,k} = 0. \quad (2.49)$$

Это равенство устанавливает связь между порождающей и проверочными матрицами [n, k]-кода, и по нему можно определить одну из них, если известна другая.

Если к какой-либо строке v_i порождающей матрицы $G_{n,k}$ прибавить линейную комбинацию других строк, то от этого она не изменится (в том смысле, что останется порождающей матрицей того же самого [n,k]-кода). Действительно, пусть строка v_i матрицы $G_{n,k}$ заменена строкой v_i', являющейся суммой строки v_i и линейной комбинации других строк:

$$v_i' = a_1v_1 + a_2v_2 + ... + v_i + ... + a_kv_k. \quad (2.50)$$

Посмотрим, какое влияние оказывает такая модификация порождающей матрицы на общий вид разрешенной кодовой комбинации. Запишем разрешенную кодовую комбинацию в общем виде:

$$v = g_1v_1 + g_2v_2 + ... + g_iv_i + ... + g_kv_k, \quad (2.51)$$

где $g_i \in \{0, 1\}$; $i = 1, 2, ..., k$.

После замены строки v_i на строку v_i' получаем:

$$v = g_1v_1 + g_2v_2 + ... + g_i(a_1v_1 + a_2v_2 + ... + v_i + ... + a_kv_k) + ... + g_kv_k = \\ = (g_1 + g_ia_1)v_1 + (g_2 + g_ia_2)v_2 + ... + g_iv_i + ... + (g_k + g_ia_k)v_k. \quad (2.52)$$

Так как коэффициенты $(g_j + g_ia_j) \in \{0, 1\}$, $j = 1, 2, ..., k$, то общий вид разрешенной кодовой комбинации не изменился и матрица с новой строкой v_i' порождает тот же самый код. Отсюда следует, что при сложении попарно, по три, по четыре и т. д. кодовых комбинаций и записи полученной суммы на место одного из слагаемых изменяется лишь вид матрицы, но не затрагивается ее сущ-

ность, т. е. матрица, претерпевшая указанную операцию, порождает тот же набор разрешенных кодовых комбинаций, что и исходная матрица.

Путем замены строк матрицы $G_{n,k}$ в соответствии с формулой (2.52) можно от любого ее вида перейти к каноническому виду:

$$G_{n,k} = \begin{vmatrix} 1 & 0 & \dots & 0 & b_{11} & b_{12} & \dots & b_{1r} \\ 0 & 1 & \dots & 0 & b_{21} & b_{22} & \dots & b_{2r} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & b_{k1} & b_{k2} & \dots & b_{kr} \end{vmatrix}. \qquad (2.53)$$

Здесь информационные разряды в порождающей матрице канонического вида представлены единичной подматрицей.

Пусть подматрица информационных разрядов порождающей матрицы имеет вид:

$$A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \dots & \dots & \dots & \dots \\ a_{k1} & a_{k2} & \dots & a_{kk} \end{vmatrix}. \qquad (2.54)$$

Здесь все вектор-строки линейно независимы. Предположим, что это не так и существует строка v_i, которая может быть выражена в виде линейной комбинации остальных строк. Представим строку v_i через сумму других строк и запишем v_i на место какой-либо строки, входящей слагаемым в данную сумму. Таким образом, если верно наше предположение, то в подматрице A могут существовать две одинаковые строки, что, в свою очередь, влечет за собой равенство соответствующих проверочных разрядов, так как должно выполняться равенство (2.36). В результате получаем, что наличие в подматрице A информационных разрядов линейно зависимых строк влечет за собой наличие в порождающей матрице линейно-зависимых строк. Но так как, по определению, в порождающей матрице все строки независимы, то и в подматрице A этой матрицы не может существовать линейно зависимых строк.

Линейно независимые вектор-строки матрицы A порождают k-мерное линейное пространство, содержащее 2^k элементов, которое может быть выражено любой системой k линейно независимых векторов, принадлежащих этому пространству, в том числе и единичной матрицей размера $k \times k$:

$$\begin{vmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{vmatrix}. \qquad (2.55)$$

Канонический вид порождающей матрицы удобен тем, что существует простая связь между элементами порождающей и проверочной матриц: для определения проверочной матрицы $[n, k]$ -кода, порождаемого матрицей вида (2.53), нужно транспонировать подматрицу проверочных разрядов матрицы $G_{n,k}$ и приписать справа к полученной матрице единичную матрицу размерности $r \times r$. Таким образом, матрице (2.53) соответствует проверочная матрица (2.45).

Коды Хэмминга. К систематическим корректирующим $[n, k]$-кодам относятся коды Хэмминга с кодовым расстоянием $d = 3$. Из табл. 2.2 видно, что число разрешенных кодовых комбинаций для кодов с $d = 3$ равно $N \le 2^n(1 + n)^{-1}$. Для кодов Хэмминга выбрано предельное значение разрешенных кодовых комбинаций $N = 2^n(1 + n)^{-1}$, а число информационных разрядов k определяется как:

$$k = \log[2^n(1 + n)^{-1}] = n - \log(n + 1). \qquad (2.56)$$

Данное уравнение имеет целочисленные решения $k = 0,1,4,11,26,...$, которые и определяют соответствующие коды Хэмминга [3,1]-код, [7,4]-код, [15,11]-код и т. д.

Рассмотрим, в качестве примера, построение [7,4]-кода. Для этого воспользуемся каноническим представлением (2.53) порождающей матрицы $G_{n,k}$. Подматрица проверочных разрядов этой матрицы должна состоять из различных ненулевых строк. Определим зависимость числа проверочных r и информационных k разрядов для кодов Хэмминга:

$$2^k = 2^n(1 + n)^{-1} \text{ или } 2^k(1 + n) = 2^k 2^r,$$

учитывая, что $n = k + r$, получим $k + r + 1 = 2^r$ или $k = 2^r - r - 1$.

Общее число ненулевых строк, которые можно составить из r-разрядной кодовой комбинации равно $2^r - 1$, из них r следует вычесть для образования единичной матрицы $r \times r$, которую следует добавить к транспонированной матрице проверочных разрядов при определении проверочной матрицы $H^{\mathrm{T}}_{n,k}$. Тогда остается $(2^r - r - 1)$ r-разрядных строк с числом единиц не менее двух. Это число равно числу информационных разрядов k. Распределение ненулевых строк для [7,4]-кода представлено на рис. 2.14. В этом случае порождающая матрица будет содержать единичную подматрицу размерностью $k \times k$ и подматрицу проверочных разрядов размерностью $k \times r$:

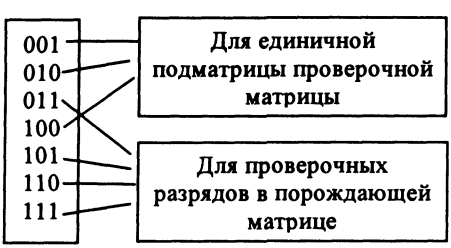


Рис. 2.14. Распределение ненулевых строк [7,4]-кода

$$G_{7,4} = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{vmatrix}. \qquad (2.57)$$

Проверочная матрица строится путем транспонирования подматрицы проверочных разрядов порождающей матрицы и добавления единичной подматрицы $r \times r$:

$$H_{7,4} = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{vmatrix}. \qquad (2.58)$$

Из этой проверочной матрицы легко получить систему проверок, связывающую проверочные и информационные разряды кода по формуле (2.36). Хэмминг предложил размещать проверочные разряды в позициях кодовой комбинации, кратных целой степени двойки, это позволяет по виду синдрома сразу определять ошибочный разряд кода. Рассмотрим алгоритмы кодирования и декодирования на примере [7,4]-кода Хэмминга.

Алгоритм кодирования. Все номера позиций кода нумеруют в двоичной системе счисления, начиная с единицы p-разрядным двоичным числом: $p = \lceil \log n \rceil$, где $\lceil\ \rceil$ – ближайшее большее целое, n – число разрядов кода $c_n c_{n-1} \ldots c_j \ldots c_1$.

Проверочные разряды размещают в позициях кода, кратных целой степени двойки 2^0, 2^1, ... и т. д.: $c_j = b_j$, $j = 2^i$, $i = 0, 1, \ldots, (r-1)$, где r – число проверочных разрядов. Значение c_j проверочного разряда определяется как сумма по mod2 тех разрядов кода, в номере которых двоичный разряд с i-м весом равен единице.

Пример. Пусть информационный кодовый вектор $v = 1101$. В коде Хэмминга этот вектор, начиная с младшего разряда, будет занимать позиции c_3, c_5, c_6 и c_7, а позиции c_1, c_2, c_4 отводятся под проверочные разряды кода. Пронумеруем все позиции кода в двоичной системе счисления: $c_{111} c_{110} c_{101} [c_{100}] c_{011} [c_{010}] [c_{001}]$ и выделим позиции для размещения проверочных разрядов:

c_{111}	c_{110}	c_{101}	c_{100}	c_{011}	c_{010}	c_{001}
1	1	0	0	1	1	0

Определим значения проверочных разрядов кода суммированием по mod2 тех разрядов кода, в номере которых двоичный разряд с i-м весом равен единице:

$$\begin{aligned} &i = 0 \to 2^0 \to c_{001} = c_{011} \oplus c_{101} \oplus c_{111}; \\ &i = 1 \to 2^1 \to c_{010} = c_{011} \oplus c_{110} \oplus c_{111}; \\ &i = 2 \to 2^2 \to c_{100} = c_{101} \oplus c_{110} \oplus c_{111}. \end{aligned} \qquad (2.59)$$

Таким образом, получен кодовый вектор $v' = 1100110$, который передают по каналу, подверженному влиянию помех. Пусть вектор ошибки равен $e = 0000100$, тогда принятая из дискретного канала кодовая комбинация будет иметь вид:

$$v'' = 1100010 = v' \oplus e. \tag{2.60}$$

Алгоритм декодирования. Вычислим значение синдрома ошибки:

$$E_{ош} = \| h_r h_{r-1} ... h_l ... h_1 \|. \tag{2.61}$$

Значение l-го разряда синдрома определяется как сумма по mod2 тех разрядов принятого кода, включая проверочные, в номере которых вес двоичного разряда совпадает с весом разряда синдрома:

$$\begin{aligned} h_1 &= c_{001} \oplus c_{011} \oplus c_{101} \oplus c_{111}; \\ h_2 &= c_{010} \oplus c_{011} \oplus c_{110} \oplus c_{111}; \\ h_3 &= c_{100} \oplus c_{101} \oplus c_{110} \oplus c_{111}. \end{aligned} \tag{2.62}$$

Для нашего примера $v'' = 1100010$:

$$\begin{aligned} h_1 &= c_1 \oplus c_3 \oplus c_5 \oplus c_7 = 0 \oplus 0 \oplus 0 \oplus 1 = 1; \\ h_2 &= c_2 \oplus c_3 \oplus c_6 \oplus c_7 = 1 \oplus 0 \oplus 1 \oplus 1 = 1; \\ h_3 &= c_4 \oplus c_5 \oplus c_6 \oplus c_7 = 0 \oplus 0 \oplus 1 \oplus 1 = 0. \end{aligned} \tag{2.63}$$

Синдром ошибки $E_{ош} = \| h_3 h_2 h_1 \| = \|011\|$ определяет в двоичной системе номер разряда, в котором обнаружена однократная ошибка. Для исправления ошибки необходимо инвертировать третий разряд – c_3 кодового вектора: $v'' = 1100110$, откуда, выделяя информационные разряды, получаем исходный кодовый вектор $v = 1101$.

Циклические коды. Линейный $[n, k]$-код называют циклическим, если в результате любого циклического сдвига кодового вектора получают другой кодовый вектор, т. е. если $v = a_0 a_1 a_2 ... a_{n-1}$ – кодовый вектор, то $v' = a_{n-1} a_0 a_1 ... a_{n-2}$ – другой кодовый вектор.

Представим кодовый вектор $v = a_0 a_1 a_2 ... a_{n-1}$ полиномом степени $(n - 1)$ или меньшей степени: $v(x) = a_{n-1} x^{n-1} + ... + a_2 x^2 + a_1 x + a_0$ – кодовый полином. Рассмотрим *двоичные циклические коды*, в которых основание x выбирают равным двум, а операция суммирования ведется по mod2. Для кодирования циклическим кодом используются так называемые порождающие полиномы $(n - k)$-степени. Пусть необходимо закодировать сообщение, представленное в виде кодового полинома $m(x) = m_{k-1} x^{k-1} + ... + m_1 x + m_0$, где k – количество информационных разрядов. Умножив $m(x)$ на x^{n-k}, получим полином степени $(n - k)$ или меньшей степени:

$$x^{n-k} m(x) = m_{k-1} x^{n-1} + ... + m_1 x^{n-k+1} + m_0 x^{n-k}. \tag{2.64}$$

Разделим (2.64) на порождающий полином $g(x)$ степени $(n-k)$:

$$x^{n-k}m(x) = q(x)g(x) + \rho(x), \qquad (2.65)$$

где $q(x)$ – частное, $\rho(x)$ – остаток.

Так как степень полинома $g(x)$ равна $(n-k)$, то степень остатка должна быть равна $(n-k-1)$ или быть меньше:

$$\rho(x) = \rho_{n-k-1}x^{n-k-1} + \ldots + \rho_2 x^2 + \rho_1 x + \rho_0. \qquad (2.69)$$

Формулу (2.65) можно представить в ином виде, где левая часть уравнения кратна $g(x)$:

$$\rho(x) + x^{n-k}m(x) = q(x)g(x). \qquad (2.70)$$

Раскроем (2.70) с учетом (2.64) в виде полинома степени $(n-1)$:

$$\begin{aligned} x^{n-k}m(x) + \rho(x) = m_{k-1}x^{n-1} + \ldots + m_1 x^{n-k+1} + m_0 x^{n-k} + \\ + \rho_{n-k-1}x^{n-k-1} + \ldots + \rho_1 x + \rho_0. \end{aligned} \qquad (2.71)$$

В данном виде формула (2.71) соответствует кодовому вектору $v = m_{k-1}\ldots m_1 m_0 \rho_{n-k-1}\ldots \rho_2\rho_1\rho_1$. Отсюда следует, что циклический код информационного сообщения $m(x)$ состоит из неизменного сообщения $m_{k-1}\ldots m_1 m_0$ и присоединенного к нему остатка $\rho_{n-k-1}\ldots \rho_1\rho_0$.

Важным свойством циклического $[n, k]$-кода является то, что порождающий полином $g(x)$ является делителем для полинома $x^n + 1$: $x^n + 1 = g(x)h(x)$, где $h(x)$ – проверочный полином. Зная $h(x)$ можно однозначно определить порождающий полином $g(x)$. Например, для циклического кода [7,4]

$$g(x) = 1 + x + x^3, \quad h(x) = \frac{x^7+1}{x^3+x+1} = x^4 + x^2 + x + 1.$$

Существует теорема, которая доказывает, что если $g(x)$ – полином степени $[n, k]$ является делителем для полинома $x^n + 1$, то $g(x)$ порождает циклический $[n, k]$-код. Действительно, любой делитель $x^n + 1$ степени $(n-k)$ может порождать циклический код. Для больших n выражение $x^n + 1$ может иметь много делителей степени $(n-k)$. Некоторые из них порождают эффективные коды, а некоторые – нет.

Обычно в качестве порождающего полинома выбирают *примитивный* полином степени $r = (n-k)$ из числа *неприводимых* полиномов той же степени. Полином $p(x)$ степени r с коэффициентами, принимающими значение из множества $\{0,1\}$, является *неприводимым*, если $p(x)$ не делится ни на один полином с двоичными положительными коэффициентами степени, меньшей чем r (делятся только на единицу и на самого себя).

Неприводимый полином $p(x)$ степени r называют *примитивным* тогда и только тогда, когда $x^n + 1$ не делят на $p(x)$ для $n < 2^r - 1$, т. е.:

$$x^n + 1 = g(x)h(x) \text{ для } n = 2^r - 1; \quad (2.72)$$

$$x^n + 1 \neq g(x)h(x) \text{ для } n < 2^r - 1. \quad (2.73)$$

Примеры примитивных полиномов :

$r = 3$ $x^3 + x + 1$; $r = 4$ $x^4 + x + 1$; $r = 5$ $x^5 + x^2 + 1$;

$r = 16$ $x^{16} + x^{12} + x^3 + x + 1$.

Процедуру кодирования циклическим кодом можно разбить на три шага:

• умножить исходный кодовый полином $m(x) = m_{k-1}x^{k-1} + \ldots + m_1x + m_0$ на x^{n-k}, что соответствует сдвигу кодового вектора в сторону старших разрядов на $(n - k)$ разрядов:

$$x^{n-k}m(x) = m_{k-1}x^{n-1} + \ldots + m_1x^{n-k+1} + m_0x^{n-k};$$

• получить остаток $\rho(x)$ от деления $x^{n-k}m(x)$ на порождающий полином $g(x)$;

• выполнить операцию конкатенации полученного кодового вектора остатка $\rho(x)$ и исходного кодового вектора полинома $m(x)$: $m_{k-1}\ldots m_1m_0\,\rho_{n-k-1}\ldots\,\rho_2\rho_1\rho_0$.

Декодирование циклического кода заключается в следующем. Пусть $v(x)$ – передаваемый кодовый полином, $r(x)$ – принятый кодовый полином. Разделив $r(x)$ на порождающий полином $g(x)$, получим:

$$r(x) = g(x)\,q(x) + s(x), \quad (2.74)$$

где $q(x)$ – частное, $s(x)$ – остаток.

Если остаток равен нулю, т. е. принятый кодовый вектор кратен порождающему полиному то, следовательно, ошибки нет или она не обнаружена. Если остаток не равен нулю, то принятый кодовый вектор не является кодовым полиномом, т. е. содержит ошибку. Таким образом, ненулевой остаток определяет наличие ошибки, т. е. представляет собой ее синдром:

$$s(x) = s_{n-k-1}x^{n-k-1} + \ldots + s_1x + s_0. \quad (2.75)$$

Пусть полином вектора ошибки имеет вид:

$$e(x) = e_{n-1}x^{n-1} + \ldots + e_1x_1 + e_0, \quad (2.76)$$

тогда $r(x) = v(x) + e(x)$ или с учетом (2.74):

$$e(x) = v(x) + q(x)g(x) + s(x). \quad (2.77)$$

Так как $v(x)$ – кодовый полином, кратный $g(x)$, т. е. $v(x) = m(x)g(x)$, то :

$$e(x) = [m(x) + q(x)]g(x) + s(x). \quad (2.78)$$

Отсюда видно, что синдром $s(x)$ является остатком от деления полинома вектора ошибок $e(x)$ на порождающий полином $g(x)$. Функция декодирующего устройства заключается в оценке полинома вектора ошибки $e(x)$ по синдрому $s(x)$.

Для различных сочетаний одиночных ошибок в кодовой комбинации двоичного циклического [7,4]-кода соответствующие им синдромы представлены в табл. 2.2.

Рассмотрим пример двоичного циклического [7,4]-кода ($n = 7$, $k = 4$). Порождающим полиномом такого кода является примитивный полином степени $(n - k)$: $g(x) = x^3 + x + 1$.

Таблица 2.2. **Определение синдрома одиночной ошибки кода [7,4]**

Ошибка $e(x)$	Синдром $s(x)$	Вектор синдрома		
		s_3	s_2	s_1
x^0	x^0	0	0	1
x^1	x^1	0	1	0
x^2	x^2	1	0	0
x^3	$x+1$	0	1	1
x^4	x^2+x	1	1	0
x^5	x^2+x+1	1	1	1
x^6	x^2+1	1	0	1

Пусть необходимо закодировать кодовый вектор с $k = 4$ 1101. Представим его в виде полинома степени $k - 1$: $m(x) = x^3 + x^2 + 1$.

Кодирование. Операция кодирования состоит из трех шагов.

1. Умножим $m(x)$ на x^{n-k} : $m(x)x^3 = (x^3 + x^2 + 1)x^3 = x^6 + x^5 + x^3$, что соответствует сдвигу кодового вектора в сторону старших разрядов на $(n - k)$ разряда и добавлению в освободившиеся разряды нулей: 1101000.

2. Разделим $m(x)x^3$ на $g(x)$:

$$
\begin{array}{l}
x^6+x^5 \quad + \quad x^3 \qquad \big| \underline{x^3+x+1} \\
x^6+ \quad x^4+ \quad x^3 \qquad\quad x^3+x^2+x+1 \\
\quad x^5+x^4 \\
\quad x^5+ \quad x^3+ \quad x^2 \\
\quad\quad x^4+ \quad x^3+ \quad x^2 \\
\quad\quad x^4+ \quad\quad x^2+x \\
\quad\quad\quad x^3+ \quad\quad x \\
\quad\quad\quad x^3+ \quad\quad x+1 \\
\quad\quad\quad\quad\quad 1-\text{остаток}
\end{array}
\qquad \text{или} \qquad
\begin{array}{l}
1101000 \quad \big| \underline{1011} \\
\underline{1011} \qquad\quad 1111 \\
\quad 1100 \\
\quad \underline{1011} \\
\quad 1110 \\
\quad \underline{1011} \\
\quad 1010 \\
\quad \underline{1011} \\
\quad 001-\textit{остаток}
\end{array}
$$

Таким образом, остаток $\rho(x) = \rho_0$.

3. Припишим остаток к информационным разрядам:

$$v(x) = m(x)x^{n-k} + \rho(x) = x^6 + x^5 + x^3 + 1.$$

или выполняем операцию конкатенации исходного кодового вектора и вектора остатка: 1101.001, в результате получаем циклический [7,4]-код.

Декодирование. Пусть вектор ошибки равен $e(x) = x^4$, тогда принятый полином будет иметь вид:

$$r(x) = v(x) + e(x) = x^6 + x^5 + x^4 + x^3 + 1 \text{ или } 1101001+0010000=1111001.$$

Для обнаружения ошибки необходимо разделить принятый полином на порождающий:

$$\begin{array}{llll} x^6+x^5+x^4+x^3+ & 1 & x^3+x+1 & \\ x^6+\quad x^4+x^3 & & x^3+x^2+1 & \\ \quad x^5 & & & \\ x^5+\quad x^3+x^2 & & & \\ \quad x^3+x^2 & +1 & & \\ \quad x^3 \quad +x & +1 & & \\ \quad x^2+x-\textit{синдром} & & & \end{array}$$

$$\begin{array}{ll} 1111001 & \underline{1011} \\ \underline{1011} & 1101 \\ 1000 & \\ \text{или}\ \underline{1011} & \\ 1101 & \\ \underline{1011} & \\ 110-\textit{вектор синдрома} & \end{array}$$

Из таблицы 2.2 по виду синдрома определяем место ошибки – разряд с весом 4.

Эффективность циклического кода. Так как порождающий полином $g(x)$ имеет степень $(n - k)$, то существует кодовый вектор, являющийся пакетом длиной $(n - k + 1)$, т. е. содержащий $(n - k + 1)$ единиц подряд. Если полином вектора ошибки $e(x)$ представляет собой пакет длиной $(n - k)$ или меньшей, то согласно выражению $e(x) = [m(x) + q(x)]g(x) + s(x)$ синдром никогда не будет равен нулю. Это означает, что циклический $[n, k]$- код пригоден для обнаружения любого пакета ошибок длиной $(n - k)$ или меньшей, а также отдельных пакетов и большей кратности. Доля необнаруженных пакетов ошибок длиной $(n - k + 1)$ составляет $2^{-(n-k-1)}$, а длиной более $(n - k + 1)$ – $2^{-(n-k)}$. Приведенный анализ показывает, что циклические коды весьма эффективны для обнаружения ошибок, поэтому их широко применяют в системах телекоммуникации.

Логический код 4B/5B. Наряду с циклическими кодами, которые используют, как правило, на канальном и выше уровнях модели OSI, для улучшения потенциальных кодов типа AMI, NRZI или 2B1Q используют другие избыточные логические коды. Логическое кодирование должно заменять длинные последовательности бит, приводящие к постоянному потенциалу в среде передачи данных, вкраплениями единиц. Как отмечалось выше, для логического кодирования характерны два метода – избыточные коды и скрэмблирование. Например, избыточный логический код 4B/5B, используемый в технологиях FDDI и Fast Ethernet, заменяет исходные символы длиной 4 бит на символы длиной в 5 бит. Так как результирующие символы содержат избыточные биты, то общее

количество битовых комбинаций в них больше, чем в исходных. Так, в коде 4В/5В результирующие символы могут содержать 32 битовых комбинации, в то время как исходные символы – только 16. Поэтому в результирующем коде можно отобрать 16 таких комбинаций, которые не содержат боль-шого количества нулей, а остальные считать *запрещенными кодовыми комбинациями*. Кроме устранения постоянной составляющей и придания коду свойства самосинхронизации, избыточные коды позволяют приемнику распознавать искаженные биты. Соответствие двоичного кода коду 4В/5В представлено в табл. 2.3. Код 4В/5В передается по линии с помощью физического кодирования по одному из методов потенциального кодирования, чувствительному только к длинным последовательностям нулей. Символы кода 4В/5В длиной 5 бит гарантируют, что при любом их сочетании на линии не могут встретиться более трех нулей подряд. Буква В в названии кода означает, что элементарный сигнал имеет 2 состояния (от английского binary – двоичный). Существуют коды и с тремя состояниями сигнала, например, в коде 8В/6Т для кодирования 8 бит исходной информации используется код из 6 сигналов, каждый из которых имеет три состояния. Избыточность кода 8В/6Т выше, чем у кода 4В/5В, так как на 256 исходных кодов приходится $3^6 = 729$ результирующих символов.

Таблица 2.3. **Соответствие двоичного кода коду 4В/5В**

Двоичный код	Код 4В/5В	Двоичный код	Код 4В/5В
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
1111	01111	1111	11101

Использование для перекодировки таблицы, аналогичной табл. 2.3, является простой операцией, поэтому это 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 4В/5В со скоростью 100 Мбит/с необходима тактовая частота передатчика 125 МГц. При этом спектр сигнала на линии расширяется по сравнению со случаем, когда по линии передается чистый, 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.

Скрэмблирование. Перемешивание данных скрэмблером перед передачей их в линию с помощью потенциального кода также является одним из способов логического кодирования. Методы скрэмблирования заключаются в побитном вычислении результирующего кода на основании бит исходного кода и полученных в предыдущих тактах бит результирующего кода. Например, скрэмблер может реализовывать соотношение:

$$B_i = A_i \oplus B_{i-3} \oplus B_{i-5}, \qquad (2.79)$$

где 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; $\oplus$ – операция исключающего ИЛИ (сложение по mod2).

Например, для исходной последовательности 110110000001 скрэмблер даст следующий результирующий код:

$B_1 = A_1 = 1$ (первые три цифры результирующего кода будут совпадать с исходным, так как на вход еще не поступили необходимые цифры)

$B_2 = A_2 = 1$
$B_3 = A_3 = 0$
$B_4 = A_4 \oplus B_1 = 1 \oplus 1 = 0$
$B_5 = A_5 \oplus B_2 = 1 \oplus 1 = 0$
$B_6 = A_6 \oplus B_3 \oplus B_1 = 0 \oplus 0 \oplus 1 = 1$
$B_7 = A_7 \oplus B_4 \oplus B_2 = 0 \oplus 0 \oplus 1 = 1$
$B_8 = A_8 \oplus B_5 \oplus B_3 = 0 \oplus 0 \oplus 0 = 0$
$B_9 = A_9 \oplus B_6 \oplus B_4 = 0 \oplus 1 \oplus 0 = 1$
$B_{10} = A_{10} \oplus B_7 \oplus B_5 = 0 \oplus 1 \oplus 0 = 1$
$B_{11} = A_{11} \oplus B_8 \oplus B_6 = 0 \oplus 0 \oplus 1 = 1$
$B_{12} = A_{12} \oplus B_9 \oplus B_7 = 1 \oplus 1 \oplus 1 = 1$

Таким образом, на выходе скрэмблера появится последовательность 110001101111, в которой нет шести нулей подряд, присутствовавших в исходном коде.

После получения результирующей последовательности приемник передает ее дескрэмблеру, где восстановится исходная последовательность на основании обратного соотношения:

$$C_i = B_i \oplus B_{i-3} \oplus B_{i-5} = (A_i \oplus B_{i-3} \oplus B_{i-5}) \oplus B_{i-3} \oplus B_{i-5} = A_i. \qquad (2.80)$$

Различные алгоритмы скрэмблирования отличаются количеством слагаемых, которые определяют цифру результирующего кода, и сдвигом между слагаемыми. Так, в сетях ISDN при передаче данных от сети к абоненту исполь-

зуется преобразование со сдвигами 5 и 23 позиции, а при передаче данных от абонента в сеть – со сдвигами 18 и 23 позиции.

Протоколы коррекции ошибок. Протоколы коррекции ошибок как правило сочетают применение циклического кодирования с решающей обратной связью. Суть этого метода состоит в следующем. Вся «полезная» информация разбивается на «порции» – кадры. Передача каждого кадра завершается передачей специальной контрольной последовательности кадра, подсчитанной по некоему, заранее определенному алгоритму. Этот рекуррентный алгоритм в процессе выдачи кадра модифицирует контрольную последовательность с помощью очередного выдаваемого байта. Принимающая сторона также подсчитывает контрольную последовательность по известному алгоритму. По окончании приема кадра проводится сравнение подсчитанной контрольной последовательности с принятым в конце кадра ее значением. По результатам сравнения приемник решает, отсюда и название «решающая обратная связь»: принимать данный кадр или его следует повторить. Результат решения этого вопроса приемник сообщает передатчику посредством некоей «квитанции». Отсюда другое название метода: «метод автоматического повтора запроса» (ARQ – Automatic Repeat reQuest).

Основную роль в обнаружении ошибок играет алгоритм вычисления контрольной последовательности кадра. Здесь использован метод циклического избыточного контроля (CRC – Cyclic Redundancy Check), определяющее свойство которого – простота кодирования: рекуррентный характер алгоритма при минимальном расходе вычислительных ресурсов. Существуют по крайней мере два алгоритма, дающих идентичный результат – битовый, модификация результата в котором проводится по каждому биту (его удобно реализовывать на аппаратном уровне с помощью сдвигового регистра) и байтово-табличный, в котором модификация результата проводится после приема/передачи целого байта (этот алгоритм больше подходит для реализации на программном уровне, поскольку требует некоторого объема памяти для хранения таблиц).

Принципы циклического помехоустойчивого кодирования с решающей обратной связью положены в основу всех аппаратных и программных реализаций наиболее широко распространенных протоколов коррекции ошибок фирмы Microcom MNP2/MNP3 и V.42 ITU-T.

Появление протоколов фирмы Microcom предшествовало выходу «Синей Книги» ITU-T с Рекомендацией V.42, в которой обобщены все достижения промышленных стандартов в этой области. Протоколы MNP (Microcom Networking Protocol) – MNP2 и MNP3 являются соответственно байт-ориентированным и бит-ориентированным протоколами.

Протокол коррекции ошибок определяет формат кадра, перечень допустимых типов кадров, логическую структуру кадра каждого типа и собственно протокол, т. е. порядок установки режима коррекции ошибок, выхода из режима и допустимого чередования кадров.

Протоколы MNP2 и MNP3. Протокол коррекции ошибок MNP2 представляет собой асинхронный байт-ориентированный протокол. Каждый элемент кадра – байт – состоит из 8 информационных бит и передается по каналу последовательно, начиная с младшего бита; выдача первого бита предваряется стартовым битом, служащим синхросигналом приемнику; после передачи последнего бита выдается стоповый бит. Если следующий байт не готов к выдаче в линию, то осуществляется передача потока стоповых битов. Таким образом, можно считать, что байт состоит как минимум из 10 бит, включая один стартовый и один стоповый биты. Отсюда вытекает два важных следствия. Во-первых, процедура входа в протокол прозрачна и не требует специального синхронного переключения обоих модемов в какой-то специфический режим работы асинхронно-синхронного преобразования данных. В любой момент модем может начать передачу символов, являющихся служебным полем кадра протокола MNP2, лишь бы приемник был готов на логическом уровне их идентифицировать. Во-вторых, реализация протокола может быть вынесена на уровень программного обеспечения компьютера.

Формат кадра MNP2 имеет следующий вид:

управляющее поле начального флага, включающее три байта: SYN, DLE и STX (16h, 10h, 02h – 16-ричные коды указанных байт);

прозрачные пользовательские данные переменной длины;

управляющее поле конечного флага, включающее 2 байта: DLE и ETX (10h, 03h);

двухбайтовая контрольная последовательность кадра, подсчитанная с помощью образующего полинома $x^{16} + x^{15} + x^2 + 1$.

Кодовая прозрачность управляющих полей обеспечивается байтом DLE, сигнализирующим о специальном значении следующего за ним байта. Если же этот байт встречается в пользовательских данных, то он должен дублироваться, чем обеспечивается прозрачность самих пользовательских данных (процедуру вставки байта DLE в пользовательские данные называют *байтстаффингом*). Поскольку протокол MNP2 – байт-ориентированный, в нем нет специального межкадрового заполнителя. Им служит межбайтовый заполнитель – поток стоповых битов. В протоколе MNP2 существуют 6 типов кадров: LR, LD, LT, LA, LN и LNA. Каждый тип кадра в поле прозрачных пользовательских данных имеет свою собственную логическую структуру, в которой кодируется признак типа кадра, а также присущие ему параметры и пользовательская информация.

Протокол коррекции ошибок MNP3 представляет собой синхронный бит-ориентированный протокол. Его кадровый формат радикальным образом отличается от MNP2 и полностью соответствует основной части Рекомендации V.42, включая асинхронно-синхронное преобразование байта, подсчет двухбайтовой контрольной последовательности кадра с точностью до образующего полинома, обеспечение прозрачности данных и межкадровый заполнитель. Все же остальное – перечень типов кадров, их логическая структура и собственно протокол – полностью идентично протоколу MNP2.

Несмотря на снижение накладных расходов, обусловленное переходом на синхронный кадровый формат, MNP3 не дает экономии вычислительных ресурсов по причине того, что процедура входа в протокол MNP3 заключается в обмене сторонами кадрами *LR* в байт-ориентированном режиме. Только согласовав с помощью этого кадра применение в дальнейшем бит-ориентированного режима, стороны синхронно в него переключаются. Таким образом, все вычислительные процедуры, присущие MNP2 (формирование кадра специфического формата, вычисление контрольной последовательности по специфическому образующему полиному, байтстаффинг и пр.) необходимо реализовывать для установки протокола MNP3.

Протокол V.42. Протокол коррекции ошибок V.42 является подмножеством, называемым LAPM (Link Access Procedure for Modems), бит-ориентированных протоколов типа HDLC (High-level Data Link Control). В отличие от асинхронного кадрового формата MNP2 формат кадра LAPM синхронный.

Кадр LAPM состоит из нескольких полей, каждое из которых включает целое число байт. Все байты в кадре передаются последовательно друг за другом без служебных битов: вслед за старшим битом предыдущего байта передается младший бит следующего. Все кадры начинаются и заканчиваются уникальной битовой последовательностью, называемой флагом – шестью единицами подряд, окаймленными нулями (01111110, 7Eh). Кодовая прозрачность тела кадра обеспечивается вставкой нулевого бита вслед за пятью единицами подряд, независимо от значения следующего бита (*битстаффинг*). Межкадровым заполнителем служит флаговая последовательность. Завершающий флаг одного кадра может одновременно служить начальным флагом следующего. Таким образом, обнаружение флага в потоке данных говорит приемнику об окончании принимаемого кадра; а появление в потоке флаговых комбинаций последовательности битов, отличных от флага, говорит о начале следующего кадра.

Формат кадра LAPM имеет следующий вид:

- начальный флаг (7Eh);
- поле адреса;
- управляющее поле;
- информационное поле;
- двухбайтовая или 4-байтовая контрольная последовательность кадра;
- конечный флаг (7Eh).

Управляющее поле кадра идентифицирует один из трех форматов кадра. Информационные кадры (*I*-формат) предназначены для передачи информации с возможностью одновременного подтверждения принятой информации, супервизорные кадры (*S*-формат) – для подтверждения принятой информации, запроса на повторную передачу или сообщения оппоненту о неготовности к приему, а ненумерованные кадры (*U*-формат) выполняют дополнительные управляющие сеансом процедуры, такие, как: установка/прекращение работы протокола, согласование параметров протокола, тестирование канала и пр. Всего в протоколе LAPM насчитывается 13 типов кадров:

1 тип кадра *I*-формата;

4 типа кадра *S*-формата: RR, RNR, REJ и SREJ;

8 типов кадров U-формата: SABME, DM, UI, DISC, UA, FRMR, XID и TEST.

Двухбайтовая контрольная последовательность кадра подсчитывается с помощью образующего полинома $x^{16} + x^{12} + x^5 + 1$, отличного от того, который используется в протоколе MNP2. Четырехбайтовая контрольная последовательность кадра подсчитывается с помощью образующего полинома

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1.$$

Выбор CRC-16 или CRC-32 проводится в процессе согласования параметров протокола с помощью кадров XID.

Вход в протокол – операция ответственная и потому ее необходимо тщательно планировать. Вызывающий модем начинает установку протокола непрерывной передачей своему оппоненту двухбайтовых «шаблонов обнаружения вызывающего» (ODP – Originator Detection Pattern) в байт-ориентированном режиме, соответствующем Рекомендации V.14 ITU-T. ODP состоит из байтов $11h$ и $91h$, разделенных 8 – 16 стоповыми битами. Отвечающий модем, приняв подряд два ODP, начинает выдавать «шаблоны обнаружения отвечающего» (ADP – Answerer Detection Pattern) в том же байт-ориентированном режиме. ADP состоит из байтов $45h$ ('E') и $43h$ ('C'), разделенных 8–16 стоповыми битами. После выдачи десяти ADP отвечающий модем переключается в синхронный режим. Вызывающий модем, приняв подряд два ADP, прекращает передачу ODP и переключается в синхронный режим. Выдача первого кадра в синхронном режиме предваряется как минимум 16 флаговыми последовательностями, с помощью которых выдерживается пауза для гарантированного переключения обоих сторон в синхронный режим. Первым кадром, как правило, является кадр XID, с помощью которого стороны согласуют параметры протокола коррекции ошибок и сжатия.

Проведем сравнительный анализ протоколов коррекции ошибок по следующим трем критериям:

1. Минимизация накладных расходов. Совокупное преимущество V.42 по этому критерию имеет несколько составляющих. Очевидное преимущество MNP3 и V.42 перед MNP2, обусловленное переходом на синхронный кадровый формат, заключается в уменьшении объема передаваемых по каналу данных по крайней мере на 20 % вследствие отказа от передачи стартовых и стоповых битов.

Обеспечение кодовой прозрачности данных в байт-ориентированном режиме приводит к увеличению объема передаваемых данных на 100 % в худшем случае, когда вся пользовательская информация состоит из одних байтов DLE. Для синхронного кадрового формата худший случай заключается в том, что пользовательская информация состоит из одних единиц (байтов FFh), что приводит к увеличению объема передаваемых данных лишь на 20 % – вставки дополнительного нуля после каждых пяти единиц.

Накладные расходы на передачу пользовательской информации посредством кадра I протокола V.42, обусловленные структурой кадра, составляют 6 байт. Аналогичные накладные расходы для кадров LT, осуществляющих передачу пользовательской информации, для протокола MNP3 составляют 8 байт, а для протокола MNP2 – 12 байт.

При двусторонней передаче информации протоколы MNP будут либо откладывать подтверждение принятой информации, неоправданно «загромождая» буфера оппонента отправленными, но неподтвержденными кадрами, либо будут вынуждены чередовать передачу пользовательской информации с подтверждениями очередных принятых кадров, т. е. увеличивать накладные расходы на 11 байт для MNP3 и на 15 байт для MNP2 (длина кадра LA). Кадр I протокола V.42 в самой своей структуре несет функцию подтверждения принятой информации и потому дополнительных накладных расходов не требует.

2. Надежность входа в протокол. Процедура входа в любой из протоколов MNP заключается в обмене взаимодействующими сторонами кадрами LR в байт-ориентированном режиме. Переключение в синхронный кадровый формат протокола MNP3 проводится только после выдачи инициатором кадра LA (и, соответственно, его приема отвечающим), подтверждающего прием ответного кадра LR. Длина кадра LR составляет 31 байт, а кадра LA – 15 байт. Таким образом, установка протокола обусловлена безошибочным приемом 31 байт отвечающим модемом, затем 31 байт вызывающим модемом и, наконец, 15 байт вновь отвечающим модемом. В то время, как для установки протокола LAPM необходимо безошибочно передать всего лишь по 4 байт в каждую сторону – по 2 ODP/ADP соответственно. Впрочем, эти 4 байт должны перемежаться потоком стоповых бит длиной в среднем в 1,5 байт. Поэтому для корректности надо говорить о 10 байт. Очевидно, что при наличии помех (в противном случае в протоколе просто нет нужды) вероятность безошибочного приема 10 байт значительно выше, чем 31 байт и, тем более, 46 байт.

Кроме того, поток ODP/ADP включает в себя не менее 10 шаблонов, т. е. каждая пара повторяется не менее 5 раз. В то время, как в случае неудачи приема кадра LR какой-либо из сторон, обмен этими кадрами будет повторен по истечении тайм-аута лишь однажды (в некоторых реализациях дважды). Превосходство в кратности повтора процедуры еще более увеличивает разницу в вероятностях успешного входа в протокол коррекции ошибок, подчеркивая преимущество протокола LAPM над MNP.

3. Гибкость. Гибкость подразумевает следующие возможности протокола:

• раздельное согласование параметров передачи – максимальный размер кадра и размер окна – для обеих сторон. Размер окна определяет количество кадров максимального размера, которое модем может хранить в памяти, ожидая их подтверждения. Оба параметра зависят от размеров оперативной памяти модемов, участвующих в сеансе связи. Поскольку они могут иметь разный объем памяти, представляется логичным, что для каждого направления передачи согласуются свои значения этих параметров. В протоколе MNP в

процессе согласования параметров выбирается одно, наименьшее, значение для передачи в обе стороны;

• кадр XID протокола LAPM, с помощью которого проводится согласование параметров, позволяет модемам обмениваться дополнительной информацией, такой, как «*ID* (идентификатор) изготовителя». Это предоставляет возможность модемам одного и того же изготовителя расширять протокол в процессе сеанса по своему усмотрению;

• возможность повышения надежности коррекции ошибок с помощью четырехбайтовой контрольной последовательности кадра (CRC-32) в особо ответственных сеансах при условии поддержки этой возможности обоими модемами. Поддержка этой возможности необязательна;

• совмещение функции передачи пользовательской информации с функцией подтверждения принятых данных;

• селективный повтор одного неверно принятого кадра. Реализация этой возможности (тип кадра SREJ) необязательна;

• кадр *U*-формата TEST позволяет в любой момент, не прекращая передачу пользовательской информации, осуществить кольцевое тестирование канала передачи данных. Поддержка этой возможности необязательна;

• пересогласование параметров передачи в любой момент времени после установки протокола. Модем может инициировать пересогласование параметров протокола, послав кадр XID в любой момент, исходя из собственных внутренних критериев. Например, посчитав, что качество канала связи ухудшилось, он может потребовать уменьшить максимальный размер кадра или включить любую из необязательных процедур: кольцевое тестирование, например, или CRC-32. MNP позволяет согласовывать параметры единожды, при входе в протокол.

Протокол LAPM содержит задел для его расширения в будущем. В частности, наличие адресного поля открывает возможности для многоточечного соединения.

Сжатие данных. Сжатие (компрессия) данных применяют для сокращения времени их передачи. Так как на сжатие данных передающая сторона тратит дополнительное время, к которому нужно еще прибавить аналогичные затраты времени на разворачивание этих данных принимающей стороной, то выгоды от сокращения времени на передачу сжатых данных обычно бывают заметны только для низкоскоростных каналов (около 64 кбит/с). Многие программные и аппаратные средства сети способны выполнять динамическую компрессию, совмещенную с передачей данных. Статическая компрессия обеспечивает предварительное сжатие данных (например, с помощью популярных архиваторов типа ARJ, RAR, WinZip), после чего они отсылаются в сеть.

Существующие алгоритмы сжатия информации можно разделить на две большие группы:

алгоритмы сжатия без потерь: алгоритм Лемпеля-Зива (Lempel-Ziv, LZ), RLE (Run Length Encoding), кодирование Хаффмена (Huffman Encoding);

алгоритмы сжатия с потерями: JPEG (Joint Photographic Expert Group), M-JPEG, MPEG (Motion Picture Expert Group).

Алгоритм Лемпеля-Зива лежит в основе архиваторов (pkzip, arj, lha) и программ динамического сжатия дисков (Stacker, DoubleSpace). Основная идея этого алгоритма состоит в том, что второе и последующие вхождения некоторой строки символов в сообщение заменяются ссылкой на ее первое появление в сообщении. Алгоритм используется для сжатия текстов и графики.

Алгоритм сжатия без потерь RLE применяют для сжатия графики (файлы формата *PCX*) и видео. Непрерывная последовательность одинаковых символов заменяется 2 байтами. В первом байте – символ, во втором – счетчик, т. е. число, которое показывает, сколько таких символов идет подряд.

Кодирование Хаффмена состоит в замене информационных символов кодовыми последовательностями различной длины. Чем чаще используется символ, тем короче кодовая последовательность.

Алгоритм сжатия с потерями JPEG ориентирован на сжатие неподвижных изображений. Он базируется на дискретном косинусном преобразовании (ДКП) неподвижного изображения, отбрасывании малых высокочастотных компонентов получаемого спектра и последующем энтропийном сжатии полученных данных.

Алгоритм M-JPEG – используют для компрессии видео, в котором каждый отдельный кадр сжимается по методу JPEG.

Алгоритм MPEG ориентирован на обработку видео. При формировании потока данных исходят из предположения о том, что два соседних кадра в видеопоследовательности мало отличаются. Опорные кадры сжимают по методу JPEG и передают относительно редко. В основном передаются изменения между соседними кадрами.

Из приведенного краткого обзора алгоритмов сжатия очевидны два вывода:

нет алгоритма, одинаково эффективного для данных разной природы;

приведенные алгоритмы рассчитаны на сжатие данных, в которых есть последовательности одинаковых символов или одни символы встречаются чаще других.

На практике используют ряд алгоритмов сжатия, каждый из которых применим к определенному типу данных. Некоторые модемы (называемые интеллектуальными) предлагают *адаптивное сжатие*, при котором в зависимости от передаваемых данных выбирается определенный алгоритм сжатия. Рассмотрим некоторые общие алгоритмы сжатия данных.

Десятичная упаковка. Когда данные состоят только из чисел, значительную экономию можно получить путем уменьшения количества используемых на цифру бит с 7 до 4, используя простое двоичное кодирование десятичных цифр вместо кода ASCII. Просмотр таблицы 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.

Относительное кодирование. Альтернативой десятичной упаковке при передаче числовых данных с небольшими отклонениями между последовательными цифрами является передача только этих отклонений вместе с известным опорным значением. Такой метод используют, в частности, в разновидностях рассмотренного выше метода импульсно-кодовой модуляции. При дифференциальной (разностной) ИКМ (ДИКМ, Differencial PCM, DPCM) вместо кодирования отсчетов кодируются разности между соседними отсчетами. Обычно разности отсчетов меньше самих отсчетов. Адаптивная ДИКМ (АДИКМ, Adaptive Differencial PCM, ADPCM) – система ДИКМ с адаптацией квантователя (АЦП и ЦАП) и предсказателя. При АДИКМ оцифровывается не сам сигнал, а его отклонение от предсказанного значения.

Символьное подавление. Часто передаваемые данные содержат большое количество повторяющихся байт. Например, при передаче черно-белого изображения черные поверхности будут порождать большое количество нулевых значений, а максимально освещенные участки изображения – большое количество байт, состоящих из всех единиц. Передатчик сканирует последовательность передаваемых байт и, если обнаруживает последовательность из трех или более одинаковых байт, заменяет ее специальной трехбайтовой последовательностью, в которой указывает значение байта, число его повторений, а также отмечает начало этой последовательности специальным управляющим символом.

Коды переменной длины. В этом методе кодирования используется 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*.

Одним из наиболее распространенных алгоритмов, на основе которых строятся неравномерные коды, является алгоритм Хаффмена, позволяющий строить коды автоматически, на основании известных частот появления символов. Рассмотрим его подробнее, предварительно определив критерий оценки эффективности кодирования.

Эффективное кодирование. Избыточность является одной из основных характеристик кода, это – полезное свойство, так как оно повышает помехоустойчивость кода. Однако для избыточных кодов для передачи по каналам связи требуется больше времени, кроме того, они занимают больший объем памяти при хранении информации. Большая избыточность не всегда оправдана требованиями помехоустойчивости при передаче и хранении информации. Поэтому возникает задача устранения избыточности, получившая название *эффективного кодирования*.

Определим $X = \{x_1, x_2, ..., x_r\}$ – входной алфавит кода Γ; множество B – выходной алфавит того же кода: $B = \{0, 1, ..., m-1\}$, где m – число элементов множества B. Код Γ сопоставляет каждому символу из входного алфавита $x_i \in X$ кодовую комбинацию, составленную из n_i символов алфавита B – $\Gamma x_r = b_1 b_2 ... b_{n_i}$.

Требуется оценить минимальную среднюю длину кодовой комбинации.

Обозначим через $p(x_i)$ вероятность появления сообщения x_i, тогда энтропия сообщений $X = \{x_1, x_2, ..., x_r\}$:

$$H(X) = -\sum_{i=1}^{r} p(x_i) \log p(x_i), \tag{2.81}$$

а средняя длина кодовой комбинации:

$$n_{ср} = \sum_{i=1}^{r} n_i \, p(x_i). \tag{2.82}$$

Максимальная энтропия, которую может иметь сообщение из символов алфавита B равно

$$H_{max} = n_{ср} \log m. \tag{2.83}$$

Для обеспечения передачи информации, содержащейся в сообщениях X, с помощью кодовых комбинаций в алфавите B должно выполняться неравенство

$$H_{max} \geq H(X). \tag{2.84}$$

В случае строгого неравенства $H_{max} > H(X)$ имеет место избыточность, которую определяют через коэффициент избыточности $K_и$:

$$K_и = \frac{H_{max} - H(X)}{H_{max}}. \tag{2.85}$$

Оценим минимальную среднюю длину кодовой комбинации, при которой еще возможна передача сообщения X без потери информации. Учитывая (2.83) и (2.84), получим:

$$n_{ср} \geq \frac{H_{max}}{\log m}, \tag{2.86}$$

т. е.

$$n_{min} = \frac{H_{max}}{\log m}. \tag{2.87}$$

Выражение (2.85) для коэффициента избыточности с учетом (2.87) можно представить в следующем виде:

$$K_и = \frac{n_{ср} - n_{min}}{n_{ср}}. \tag{2.88}$$

Под эффективным кодом понимают код, коэффициент избыточности которого равен нулю, т. е. для эффективных кодов $n_{ср} = n_{min}$ или с учетом (2.81), (2.82) и (2.83):

$$\sum_{i=1}^{r} n_i p(x_i) \log m = -\sum_{i=1}^{r} p(x_i) \log p(x_i). \qquad (2.89)$$

Объединив суммы левой и правой части, получим:

$$\sum_{i=1}^{r} p(x_i)[n_i \log m + \log p(x_i)] = 0. \qquad (2.90)$$

Если $p(x_i) \neq 0$, то необходимо, чтобы

$$n_i = -\frac{\log p(x_i)}{\log m} \quad \text{для всех } i = 1, 2, \ldots, r. \qquad (2.91)$$

Однако отношение (2.91) не всегда является целым числом и, следовательно, не для любого набора сообщений $X = \{x_1, x_2, ..., x_r\}$ с заданным распределением вероятности $p(x_1), p(x_2), ..., p(x_r)$ можно построить эффективный код с минимальной избыточностью $K_и = 0$. Всегда можно обеспечить выполнение неравенства:

$$-\frac{\log p(x_i)}{\log m} \leq n_{ср} \leq \frac{\log p(x_i)}{\log m} + 1, \qquad (2.92)$$

или умножая части неравенства на $p(x_i)$ и суммируя по i, получим критерий оценки эффективности реального кода:

$$\frac{H(X)}{\log m} \leq n_{ср} \leq \frac{H(X)}{\log m} + 1. \qquad (2.93)$$

Выражение для коэффициента избыточности равномерного двоичного кода определим по формуле (2.85), в которой под максимальной энтропией H_{max} будем понимать максимальную энтропию равномерного n-разрядного кода, $n_{ср} = n$, $m = 2$. При этом максимальная энтропия $H_{max} = n_{ср} \log m = n$. Под энтропией сообщений $H(X)$ будем понимать энтропию разрешенных кодовых комбинаций, число которых обозначим через N, тогда $H(X) = \log N$.

Подставляя выражения для H_{max} и $H(X)$ в (2.85), имеем

$$K_и = (n - \log N) / n = 1 - \log N / n. \qquad (2.94)$$

Для разделимых кодов формула упрощается:

$$K_и = 1 - (\log 2^k) / n = 1 - k/n, \qquad (2.95)$$

где n – длина кодовой комбинации; k – число информационных разрядов.

Анализ формулы (2.85) показывает, что коэффициент избыточности принимает значения от 0 (отсутствие избыточности) до 1 (избыточность неограниченно велика). Коэффициент избыточности характеризует качество помехоустойчивого кода – чем меньше избыточность кода при прочих равных условиях, тем код лучше.

Алгоритм Хаффмена. Этот алгоритм применяют для построения кодов с минимальной избыточностью. Пусть $X = \{x_1, x_2, ..., x_r\}$ – входной алфавит кода $Г$, а $B\{0, 1\}$ – выходной алфавит. С каждым символом x_i связана вероятность его появления $p(x_i)$. Необходимо каждому символу алфавита X сопоставить кодовую комбинацию в алфавите B таким образом, чтобы обеспечить минимальную избыточность кода $Г$.

Поставленную задачу можно решить построением кодового дерева. Построение графа-дерева начинают с висячих вершин, которым в качестве весов назначают вероятности $p(x_i)$, $i = 1, 2, ..., r$. Висячие вершины графа упорядочивают в соответствии с их весом, что позволяет в дальнейшем уменьшить число пересечений ребер или вовсе исключить их. Само дерево строится по следующему алгоритму.

Шаг 1. Определяют число поддеревьев графа. Если оно меньше двух, то дерево построено, и на этом действие алгоритма заканчивается. Если число поддеревьев равно или больше двух, то переходят к шагу 2 (*Замечание*: в начале построения имеется r изолированных вершин графа, являющихся поддеревьями и одновременно корнями поддеревьев).

Шаг 2. Выбирают корни двух поддеревьев графа с минимальными весами и осуществляется сращивание выбранных поддеревьев с добавлением при этом одной вершины и двух ребер. Вес вновь образованной вершины определяется как сумма весов корней выбранных поддеревьев. Левому добавленному ребру приписывается вес, равный единице, правому – равный нулю. Осуществляется переход к шагу 1.

В результате образуется кодовое дерево – граф с взвешенными ребрами. Для получения кода символа x_i достаточно выписать веса ребер, составляющих путь из корня дерева в соответствующую висячую вершину.

Пример. Пусть входной алфавит содержит восемь различных символов $x_1, x_2, ..., x_8$, выходной алфавит является двоичным $B\{0,1\}$, число символов выходного алфавита $m = 2$. Известна вероятность (частота) появления каждого символа в передаваемом сообщении: $p(x_1) = 0{,}19$; $p(x_2) = p(x_3) = 0{,}16$; $p(x_4) = 0{,}15$; $p(x_5) = 0{,}12$; $p(x_6) = 0{,}11$; $p(x_7) = 0{,}09$; $p(x_8) = 0{,}02$. Необходимо разработать код и определить коэффициент избыточности кода.

Процесс построения кодового дерева содержит семь циклов последовательного выполнения шагов 1 и 2 алгоритма (рис. 2.15). Для оценки эффективности кода промежуточные вычисления занесем в табл. 2.4.

Средняя длина кодовой комбинации

$$n_{ср} = \sum_{i=1}^{8} n_i p(x_i) = 2{,}92;$$

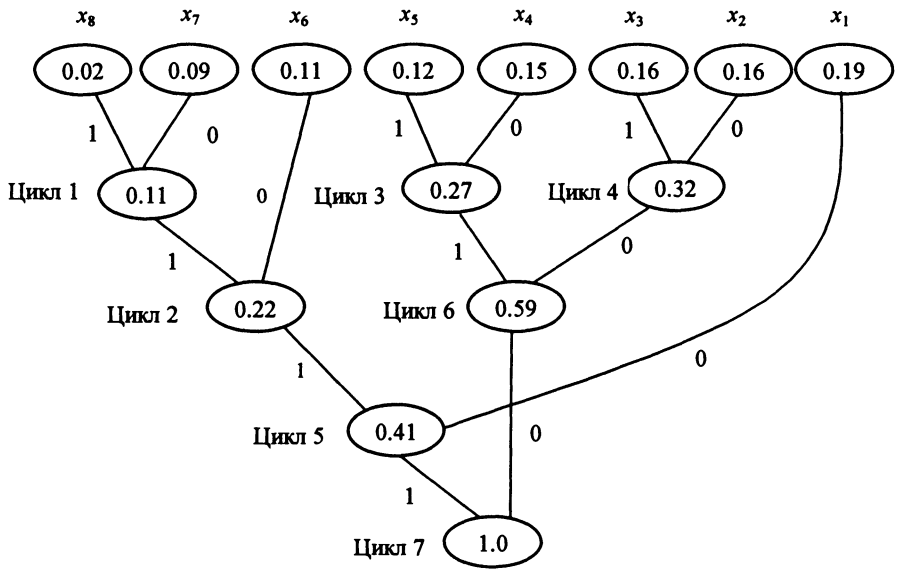


Рис. 2.15. Построение кодового дерева

энтропия сообщения, составленного из символов входного алфавита, равна

$$H(X) = -\sum_{i=1}^{8} p(x_i)\log_2 p(x_i) = 2{,}855.$$

Таблица 2.4. **Промежуточные значения вычисления энтропии для кода Хаффмена**

Символ	Кодовая комбинация	Длина кодовой комбинации (n_i)	$n_i p(x_i)$	$-p(x_i)\log_2 p(x_i)$
x_1	10	2	0.38	0.455
x_2	000	3	0.48	0.423
x_3	001	3	0.48	0.423
x_4	010	3	0.45	0.411
x_5	011	3	0.36	0.367
x_6	110	3	0.33	0.350
x_7	1110	4	0.36	0.313
x_8	1111	4	0.08	0.113

Минимальная длина кодовой комбинации выходного алфавита, необходимая для передачи сообщения, составленного из символов входного алфавита, без потери информации:

$$n_{\min} = \frac{H(X)}{\log_2 m} = \frac{2{,}855}{\log_2 2} = 2{,}855.$$

Коэффициент избыточности кода, построенного по алгоритму Хаффмена, в соответствии с (2.88) равен

$$K_{\text{и}} = \frac{n_{\text{ср}} - n_{\min}}{n_{\text{ср}}} = 0{,}022,$$

т. е. построенный код практически избыточности не имеет.

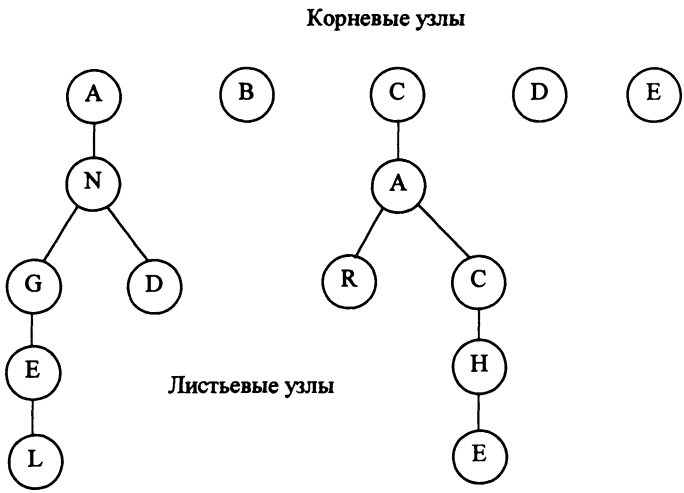


Рис. 2.16. Представление части словаря, содержащего строки: A, AN, ANGEL, AND, B, C, CA, CAR, CACHE, D, E

Наряду с рассмотренным алгоритмом Хаффмена, который требует дополнительного просмотра данных для вычисления значений $p(x_i)$, существуют его адаптивные модификации, позволяющие строить кодовое дерево «на ходу», по мере поступления данных от источника.

Протоколы сжатия MNP5 и V.42bis. В отличие от протокола MNP5 протокол V.42bis не заменяет конкретные, наиболее часто встречающиеся символы на более короткие кодовые слова, а делает это для последовательностей символов (строк). Протокол использует словарь для хранения наиболее часто встречающихся строк вместе с кодовыми словами, которые их представляют. Словарь строится и модифицируется динамически. Размер словаря может быть различным, стандартизировано только минимальное значение – 512 элементов (строк). Конкретное значение выбирается обоими модемами при установлении соединения. Кроме того, согласовывается максимальная длина строки, которая может быть сохранена в словаре, в диапазоне от 6 до 250 символов.

Словарь может быть представлен как набор деревьев, в котором корню каждого дерева соответствует символ алфавита и, наоборот, каждому символу соответствует дерево в словаре. Каждое дерево представляет набор известных (уже встретившихся) строк, начинающихся с символа, соответствующего корню. Каждый узел дерева соответствует набору строк в словаре, а каждый листьевой узел соответствует одной известной строке. Набор деревьев представленный на рис. 2.16, представляет строки A, AN, ANGEL, AND, B, C, CA, CAR, CACHE, D, E. Каждый листьевой узел не имеет подчиненных узлов и фактически соответствует последнему символу в строке. И наоборот, узел, который не имеет родительского узла, соответствует первому символу в строке. В самом начале каждое дерево в словаре состоит только из корневого узла,

которому присвоено уникальное кодовое слово. По мере поступления символов из присоединенного к модему терминала, выполняется процедура отождествления накопленной (отождествленной) к предыдущему шагу строки и текущего символа (string-matching procedure). Фактически эта процедура сводится к поиску строки, дополненной текущим символом, в словаре. Она начинается с единственного символа (и в этом случае всегда завершается успешно, так как в словаре всегда есть односимвольные строки, соответствующие каждому символу алфавита). Если отождествленная на предыдущем шаге строка плюс символ соответствует элементу словаря (найдена в нем), и этот элемент не был создан при предыдущем выполнении процедуры отождествления строки (весьма важное, принципиальное и тонкое ограничение, позволяющее приемнику поддерживать адекватное состояние словаря в некоторых частных случаях комбинаций повторяющихся подстрок во входном потоке), строка дополняется текущим символом и будет использована при следующем вызове процедуры отождествления. Процесс продолжается до тех пор, пока строка не достигнет максимально возможной длины (согласованной модемами при установлении соединения), либо дополненная строка не найдена в словаре, либо она была найдена, но этот элемент был создан при предыдущем вызове. В этом случае, присоединенный к строке символ удаляется из нее и называется «неотождествленным» (unmatched), строка кодируется кодовым словом, а на следующем шаге будет состоять только из неотождествленного символа. Во время процесса сжатия словарь динамически дополняется новыми элементами (строками), которые соответствуют подстрокам, встречающимся в потоке данных. Новые строки образуются добавлением неотождествленного символа к существующей строке, что означает создание нового узла дерева. Например, в случае, если текущая отождествленная строка *CA*, а последнее переданное кодовое слово соответствовало строке *AN*, появление символа *T* приводит к добавлению в словарь строки *CAT* (рис. 2.17). На следующем шаге текущая строка соответствует неотождествленному символу *T*.

Словари должны быть модифицированы в обоих модемах: на передающей (передатчик) и принимающей (приемник) сторонах соединения. Важно понимать, что передатчик всегда находится на один шаг (на одну строку) впереди приемника в цикле модификации словаря. Таким образом, в принимающем модеме первый символ принятого кодового слова (который будет равен C) должен быть использован для модификации словаря приемника. Приемник V.42bis всегда полагает, что первый символ каждой строки (соответствующей принятому кодовому слову) должен быть использован для дополнения предыдущей строки и создания нового элемента словаря. Состояние фрагмента словаря приемника после приема кодового слова, соответствующего CA, при том, что предыдущее кодовое слово соответствовало строке AN, показано на рис. 2.18. При приеме приемником первого символа T следующего кодового слова его словарь будет иметь вид, изображенный на рис. 2.17.

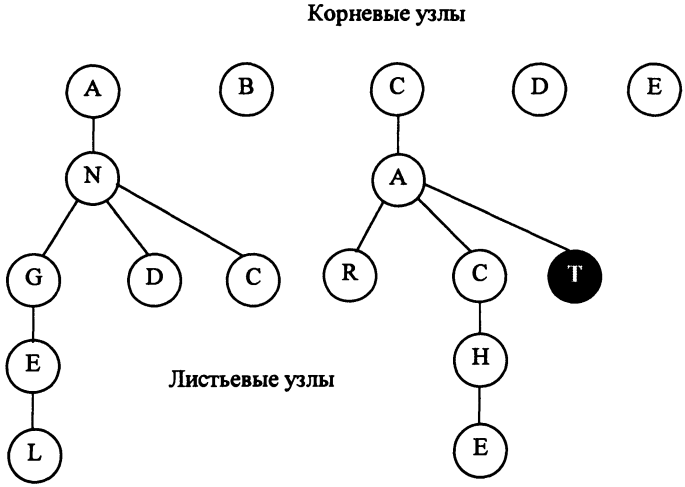


Рис. 2.17. Изменения в части словаря передатчика после последовательного получения от терминала символов A, N, C, A и T

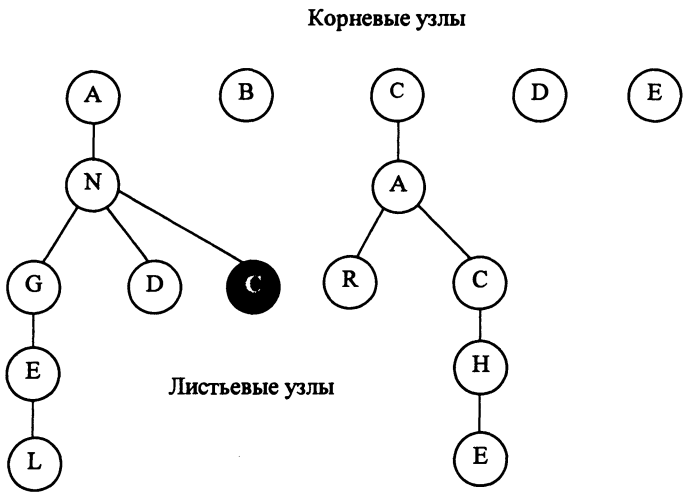


Рис. 2.18. Изменения в части словаря приемника после последовательного получения кодовых слов, соответствующих строкам AN и CA

В протоколе сжатия V.42bis реализованы и другие возможности. К наиболее важным из которых относятся следующие.

1. Определен механизм удаления элементов словаря при его переполнении. На понятийном уровне он заключается в том, чтобы после создания нового элемента удалить самый «старый» (по времени создания) листьевой элемент вне зависимости от частоты его использования. Кажущийся недостаток – не-

возможность учесть частоту появления подстрок и не удалять наиболее часто встречающиеся – частично устраняется логикой дополнения словаря: если в потоке данных часто встречаются подстроки, которые уже есть в словаре, то новые элементы словаря создаются редко и словарь медленно переполняется.

2. Реализован механизм постепенного увеличения длины кодового слова. Для представления максимального номера элемента (строки) словаря требуется 9 бит для словаря в 512 элементов, 10 – для словарей, содержащих до 1024 элементов, 11 – до 2048 элементов и т. д. Однако не все номера должны быть представлены максимальным количеством бит, и этот механизм означает, что размер кодового слова увеличивается с 9 до максимального значения по мере заполнения словаря. Это снижает накладные расходы на первоначальных этапах.

3. Существуют два режима работы передатчика: Прозрачный и режим Сжатия. Режим Сжатия описан выше, Прозрачный режим отличается от него тем, что передача кодовых слов не осуществляется, а каждый приходящий на вход передатчика символ транслируется в линию и далее в приемник. Если на вход передатчика поступают хорошо перемешанный (в статистическом смысле) поток символов, то высока вероятность, что каждый следующий символ будет «неотождествленным» (такая же ситуация складывается сразу после инициализации словаря – он еще пуст). На каждый принятый и «неотождествленный» символ на выход передается кодовое слово. Длина символа, как правило, 8 бит (здесь и далее предполагается, что символы представляют из себя октеты, хотя стандарт и допускает реализацию на нетрадиционных аппаратных средствах), минимальная длина кодового слова – 9 бит. В этом случа эффективность сжатия будет отрицательной, и потери могут составлять десятки процентов.

Сравнивая качественные и количественные характеристики протоколов сжатия V.42bis и Microcom MNP5, следует отметить, что оба алгоритма используют адаптивную технологию замены определенной входной последовательности на выходную битовую последовательность. Протокол V.42bis кодирует последовательность символов кодовым словом постепенно нарастающего и всегда большего, чем длина символа, размера. Протокол MNP5 устраняет длинные последовательности одинаковых символов конструкцией со счетчиком повторения и затем кодирует отдельные символы в соответствии с частотой их повторения кодовыми словами переменной длины. Кодовые слова могут быть короче длины символа для часто повторяющихся символов и длиннее в противном случае. Этот протокол не определяет Прозрачного режима, и следовательно возможны ситуации, приводящие к значительному расширению выходного потока. В случае корректной реализации V.42bis это практически невозможно, кроме того, V.42bis поддерживает возможность переинициализации словарей, что позволяет алгоритму лучше адаптироваться к хорошо пере-

мешанному потоку. Несомненным преимуществом протокола V.42bis является возможность параметризации протокола, что позволяет создавать более гибкие реализации.

Реальное сжатие по протоколу V.42bis на 20...30 % эффективнее, чем сжатие по MNP5 и на 5...10 % эффективнее, чем по MNP7.

2.3. Методы и технологии передачи данных

Виды связи и режимы передачи данных

При передаче данных между двумя взаимодействующими объектами возможны три вида связи:

симплексный – используется, когда передача данных должна осуществляться только в одном направлении, например в системах контроля, в которых информация с датчиков передается в управляющий компьютер через регулярные промежутки времени;

полудуплексный – применяется, когда два взаимодействующих объекта хотят обмениваться информацией поочередно, т.е. канал используется поочередно для передачи данных в обоих направлениях. Ясно, в таком режиме каждый объект должен иметь возможность переключаться от состояния передачи к состоянию приема;

дуплексный – используется для обмена данными между двумя взаимодейст-вующими объектами (устройствами) в обоих направлениях одновременно, например, когда пропускная способность канала позволяет потоку данных осуществляться в обоих направлениях независимо.

При обмене данными на физическом уровне единицей информации является бит, поэтому средства физического уровня всегда поддерживают побитовую синхронизацию между приемником и передатчиком. Чтобы приемник мог правильно декодировать и интерпретировать получаемый набор битов, он должен знать:

скорость передачи битов, определяемую интервалом времени, выделяемым на один битовый разряд;

начало и конец каждого элемента (символа или байта);

начало и конец каждого полного блока сообщения или кадра.

Эти три фактора называют соответственно побитной или тактовой синхронизацией, побайтной или посимвольной синхронизацией и поблочной или покадровой синхронизацией.

Канальный уровень оперирует кадрами данных и обеспечивает синхронизацию между приемником и передатчиком на уровне кадров. В обязанности приемника входит распознавание начала первого байта кадра, границ полей кадра и признака окончания кадра. Обычно достаточно обеспечить синхронизацию на указанных двух уровнях – битовом и кадровом, – чтобы передатчик и приемник обеспечили устойчивый обмен информацией. Однако при плохом качестве линии связи (как правило это относится к телефонным коммутируемым

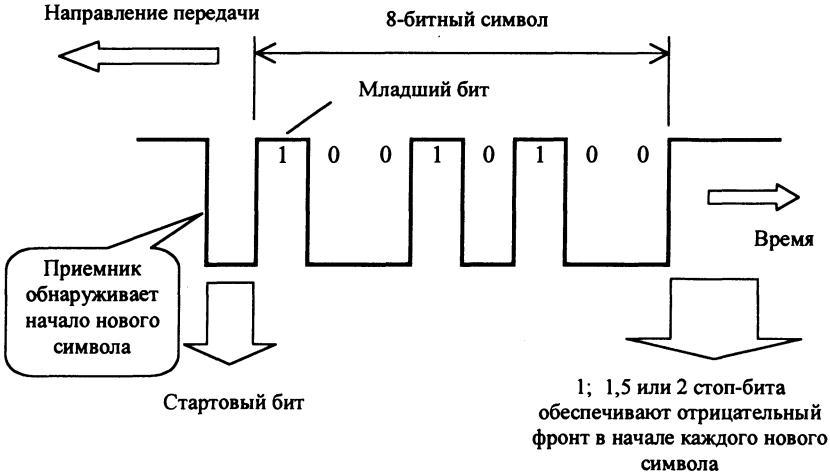


Рис. 2.19. Асинхронная передача

каналам) для удешевления аппаратуры и повышения надежности передачи данных вводят дополнительные средства синхронизации на уровне байт. Такой режим работы называется *асинхронным* или *старт-стопным.* Использование такого режима работы обусловлено наличием устройств, которые генерируют байты данных в случайные моменты времени. Так работает клавиатура дисплея или другого терминального устройства, с которого человек вводит данные для обработки их компьютером.

В асинхронном режиме каждый байт данных сопровождается специальными сигналами «старт»-стартовый бит и «стоп»-стоповый(ые) бит(ы) (рис. 2.19). Назначение этих сигналов состоит в том, чтобы, во-первых, известить приемник о приходе данных и, во-вторых, чтобы дать приемнику достаточно времени для выполнения некоторых функций, связанных с синхронизацией, до поступления следующего байта. Сигнал «старт» имеет продолжительность в один тактовый интервал, а сигнал «стоп» может длиться один, полтора или два такта, поэтому говорят, что используется один, полтора или два бита в качестве стопового сигнала, хотя эти сигналы не несут информации. Асинхронным данный режим называют потому, что каждый байт может быть несколько смещен во времени относительно побитовых тактов предыдущего байта. Такая асинхронность передачи байт не влияет на корректность принимаемых данных, так как в начале каждого байта происходит дополнительная синхронизация приемника с источником за счет стартового бита. Более «свободные» временные допуски определяют низкую стоимость оборудования асинхронной системы.

При *синхронном* режиме передачи старт-стопные биты между каждой парой байт отсутствуют и весь блок или кадр данных передается как одна цепочка битов без каких-либо задержек между 8-битными элементами. Чтобы приемник обеспечивал различные уровни синхронизации, необходимо выполнение следующих требований:

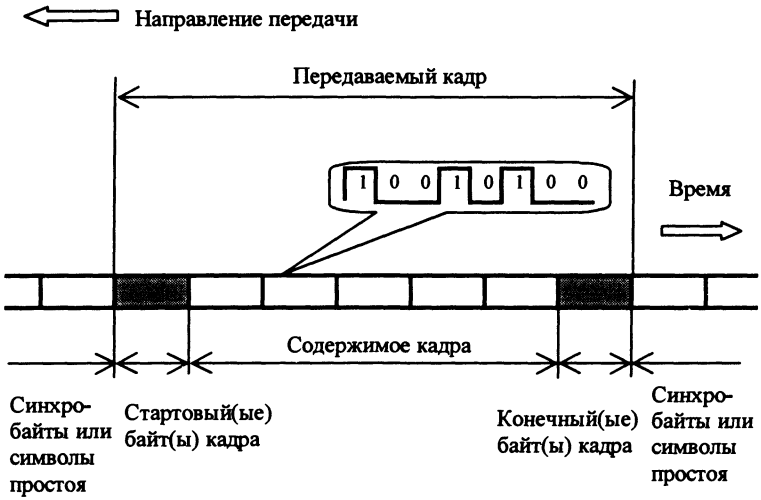


Рис. 2.20. Синхронная передача

• передаваемая цепочка битов должна быть закодирована так, чтобы приемник мог осуществлять побитовую синхронизацию;

• каждому кадру должен предшествовать один или более зарезервированных байтов или символов, благодаря чему приемник может надежно разделить полученную цепочку битов по границам байтов или символов (побайтная или посимвольная синхронизация);

• содержимое каждого кадра обрамляется парой зарезервированных байтов или символов.

Благодаря последнему требованию приемник оповещается о поступлении кадра данных и об окончании кадра (рис. 2.20). При наличии промежутков времени между передачей двух последовательных кадров в этот период либо непрерывно передаются синхробайты бездействия (простоя), что позволяет приемнику поддерживать побитную или побайтную синхронизацию, либо каждому кадру предшествует один или несколько специальных синхронизирующих байтов или символов, например 01111110, что позволяет приемнику вновь войти в байтовый синхронизм с передатчиком.

Для обеспечения побитовой синхронизации используют самосинхронизирующиеся коды.

Методы передачи данных канального уровня

Канальный уровень обеспечивает передачу пакетов данных, поступающих от протоколов верхних уровней, узлу назначения, адрес которого также указывает протокол верхнего уровня. Протоколы канального уровня оформляют переданные им пакеты в кадры собственного формата, помещая указанный адрес назначения в одно из полей такого кадра, а также сопровождая кадр контрольной суммой. Протокол канального уровня предназначен для доставки

кадров данных, как правило, в пределах сетей с простой топологией связей и однотипной или близкой технологией. Другой областью действия протоколов канального уровня являются связи типа «точка-точка» глобальных сетей, когда протокол канального уровня ответственен за доставку кадра непосредственному соседу. Адрес в этом случае не имеет принципиального значения, а на первый план выходит способность протокола восстанавливать искаженные и утерянные кадры, так как плохое качество территориальных каналов, особенно коммутируемых телефонных, часто требует выполнения подобных действий.

Основными характеристиками метода передачи, работающего на канальном уровне, являются следующие:

- асинхронный/синхронный;
- байт-ориентированный/бит-ориентированный;
- с предварительным установлением соединения/дейтаграммный;
- с обнаружением искаженных данных/без обнаружения;
- с обнаружением потерянных данных/без обнаружения;
- с восстановлением искаженных и потерянных данных/без восстановления;
- с поддержкой динамической компрессии данных/без поддержки.

Многие из них характерны не только для протоколов канального уровня, но и для протоколов более высоких уровней.

Асинхронные протоколы

Асинхронные протоколы представляют собой один из первых способов связи. Эти протоколы оперируют не с кадрами, а с отдельными символами, которые представлены байтами со старт-стоповым обрамлением.

В асинхронных протоколах применяются стандартные наборы символов, чаще всего ASCII или EBCDIC. Первые 32 или 27 кодов в этих наборах являются специальными. Они не отображаются на дисплее или принтере и используются асинхронными протоколами для управления режимом обмена данными. В самих пользовательских данных, которые представляют собой буквы, цифры, а также такие знаки, как @, %, $ и т. п., специальные символы никогда не встречаются, так что проблемы их отделения от пользовательских данных не существует.

Постепенно асинхронные протоколы усложнялись и стали наряду с отдельными символами использовать целые блоки данных, т. е. кадры. Примерами асинхронных протоколов являются популярный протокол X MODEM, который передает файлы между двумя компьютерами по асинхронному модему и протокол коррекции ошибок в модемной связи MNP2. В этих протоколах часть управляющих операций выполняется посылкой в асинхронном режиме отдельных символов, а часть данных – блоками, что более характерно для синхронных протоколов.

Байт-ориентированные и бит-ориентированные протоколы

В синхронных протоколах между пересылаемыми символами (байтами) нет стартовых и стоповых сигналов, поэтому отдельные символы в этих протоколах пересылать нельзя. Все обмены данными осуществляются кадрами, которые имеют в общем случае заголовок, поле данных и концевик. Все биты кадра передаются непрерывным синхронным потоком, что значительно ускоряет передачу данных. Так как байты в этих протоколах не отделяются друг от друга служебными сигналами, то прежде всего приемник должен распознать границы байт. Затем приемник должен найти начало и конец кадра, а также определить границы каждого поля кадра – адреса назначения, адреса источника, служебных полей заголовка, поля данных и контрольной суммы, если она имеется.

Большинство протоколов допускает использование в кадре поля данных переменной длины. Иногда и заголовок может быть переменной длины. Обычно протоколы определяют максимальное значение длины поля данных – *максимальную единицу передачи данных* (MTU – Maximum transfer Unit). В некоторых протоколах задается также минимальное значение длины поля данных. Например, протокол Ethernet требует, чтобы поле данных содержало не менее 46 байт данных (если приложение хочет отправить меньшее количество байт, то оно обязано дополнить их до 46 байт любыми значениями). Другие протоколы разрешают использовать поле данных нулевой длины, например FDDI.

Существуют протоколы с кадрами фиксированной длины, например, в сетях ATM кадры имеют фиксированный размер 53 байт, включая служебную информацию. Для таких протоколов необходимо решить только первую часть задачи – распознать начало кадра.

Синхронные протоколы канального уровня бывают двух типов: *байт-ориентированные* (иногда их называют символьно-ориентированные или знак-ориентированные) и *бит-ориентированные*. Для них характерны одни и те же методы синхронизации бит. Главное различие между ними заключается в методе синхронизации символов и кадров.

Байт-ориентированные протоколы. Эти протоколы используют в основном для передачи блоков отображаемых символов, например текстовых файлов. При синхронной передаче стоповые и стартовые биты отсутствуют. Синхронизация здесь осуществляется за счет того, что передатчик добавляет два или более управляющих символа – символы синхронизации (SYN) – перед каждым блоком символов. В коде ASCII символ SYN имеет двоичное значение 00010110 (16*h*), это несимметричное относительно начала символа значение позволяет легко разграничивать отдельные символы SYN при их последовательном приеме. Символы SYN выполняют две функции: во-первых, они обеспечивают приемнику побитную синхронизацию, во-вторых, как только битовая синхронизация достигается, они позволяют приемнику начать распознавание границ символов SYN. После того как приемник начал отделять один символ от другого, можно задавать границы начала кадра с помощью другого

специального символа. Обычно в символьных протоколах для этих целей используется символ STX (Start of TeXt)-02h. Другой символ – ETX (End of TeXt)-03h –отмечает окончание кадра.

Однако такой простой способ выделения начала и конца кадра хорошо работает только в том случае, если внутри кадра нет символов STX и ETX. При подключении к компьютеру алфавитно-цифровых терминалов такая задача действительно не возникает. Тем не менее, синхронные байт-ориентированные протоколы позднее стали использовать и для связи компьютера с компьютером, а в этом случае данные внутри кадра могут быть любые, когда, например, между компьютерами передается программа. Наиболее популярными протоколами такого типа являются протоколы IBM-2848 и BSC компании IBM. Протокол BSC работает в двух режимах – непрозрачном (некоторые специальные символы внутри кадра запрещены) и прозрачном (разрешена передача внутри кадра любых символов, в том числе и ETX). Прозрачность достигалась за счет *байтстаффинга* – перед управляющими символами STX и ETX всегда вставлялся символ DLE (Data Link Escape). Такая процедура называется вставкой символов или. Если в поле данных кадра встречается последовательность DLE ETX, то передатчик удваивает символ DLE, т. е. порождает последовательность DLE DLE ETX. Приемник, встретив подряд два символа DLE DLE, всегда удаляет первый, но оставшийся DLE уже не рассматривает как начало управляющей последовательности, т. е. оставшиеся символы DLE ETX воспринимается как пользовательские данные.

Бит-ориентированные протоколы. Потребность в паре символов в начале и конце каждого кадра вместе с дополнительными символами DLE означает, что байт-ориентированная передача не эффективна для передачи двоичных данных, так как приходится в поле данных кадра добавлять достаточно много избыточных данных. Кроме того, формат управляющих символов для разных кодировок различен, например, в коде ASCII символ SYN равен 0010110, а в коде EBCDIC – 00110010. Следовательно этот метод допустим только с определенным типом кодировки, даже если кадр содержит только двоичные данные. Чтобы это преодолеть используют более универсальный метод бит-ориентированной передачи. Этот метод в настоящее время применяется при передаче как двоичных, так и символьных данных.

На рис. 2.21 показаны схемы бит-ориентированной передачи, отличающиеся способом обозначения начала и конца каждого кадра. Схема, представленная на рис. 2.21, *а,* похожа на схему с символами STX и ETX в байт-ориентированных протоколах. Начало и конец каждого кадра отмечены одной и той же 8-битовой последовательностью – 01111110, называемой флагом. Термин «бит-ориентированный» используется потому, что принимаемый поток битов сканируется приемником на побитовой основе для обнаружения стартового флага, а затем во время приема для обнаружения стопового флага. Поэтому длина кадра в этом случае не обязательно должна быть кратна байту.

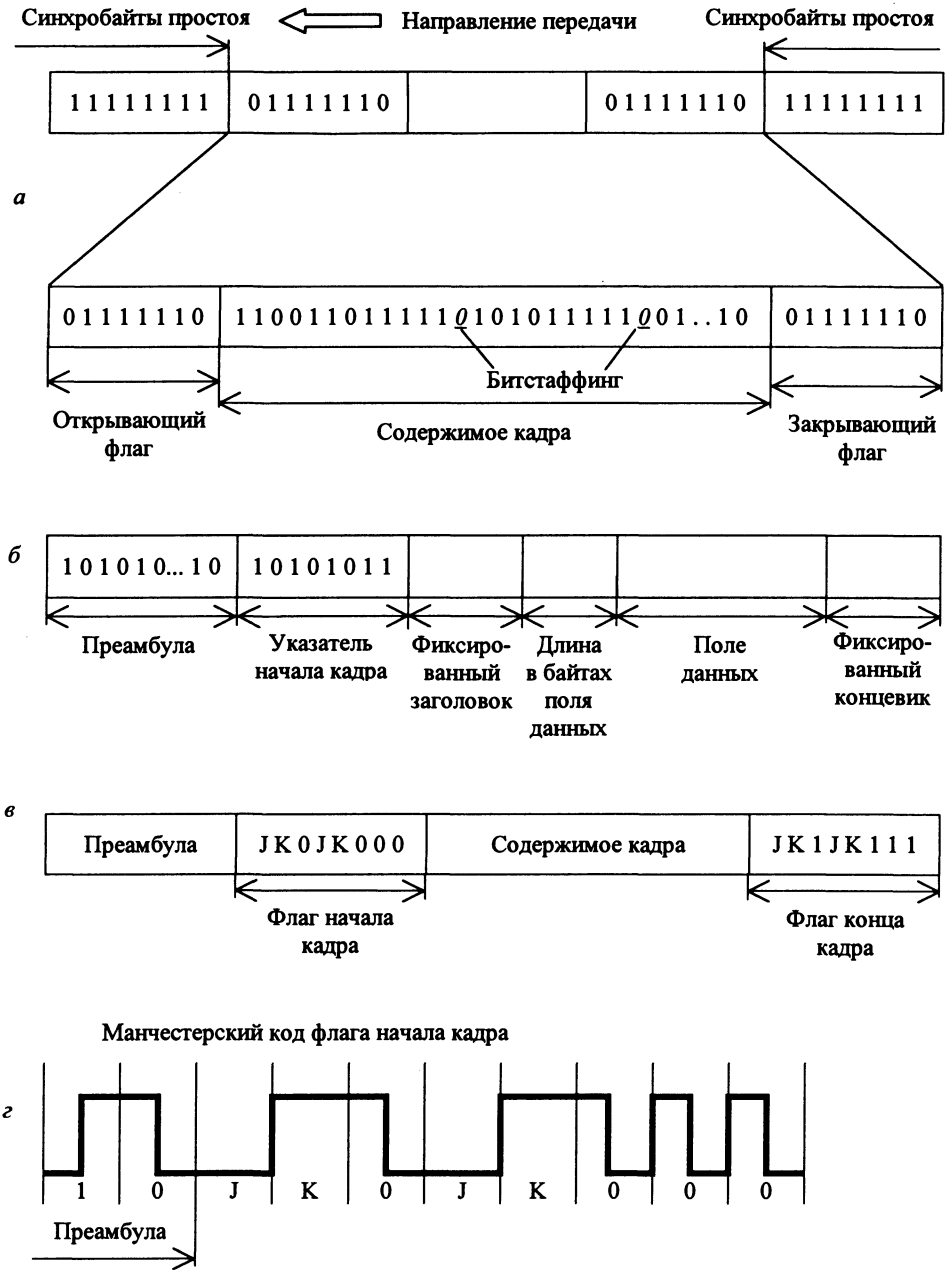


Рис. 2.21. Способы выделения начала и конца кадра в бит-ориентированных протоколах

Чтобы обеспечить синхронизацию приемника, передатчик посылает последовательность байтов простоя (11111111), предшествующую стартовому флагу. Для достижения прозрачности данных в этой схеме необходимо, чтобы флаг не присутствовал в поле данных кадра. Это достигается с помощью приема, известного как вставка 0-го бита, – *битстаффинга.* Схема вставки бита работает только во время передачи поля данных кадра. Если эта схема обнаруживает, что подряд передано пять единиц, то она в любом случае автоматически вставляет дополнительный ноль. Поэтому последовательность 01111110 никогда не появится в поле данных кадра. Аналогичная схема работает в приемнике и выполняет обратную функцию. Когда после пяти единиц обнаруживается ноль, он автоматически удаляется из поля данных кадра. Битстаффинг экономичнее байтстаффинга, так как вместо лишнего байта вставляется один бит, следовательно, скорость передачи пользовательских данных в этом случае снижается медленнее.

Во второй схеме (см. рис. 2.21, *б)* для обозначения начала кадра предусмотрен только стартовый флаг, а для определения конца кадра используется поле длины кадра, которое при фиксированных размерах заголовка и концевика чаще всего имеет смысл длины поля данных кадра. Эта схема наиболее применима в локальных сетях, в которых для обозначения факта незанятости среды вообще не передается никаких символов. Чтобы все остальные станции вошли в битовую синхронизацию, посылающая станция предваряет содержимое кадра последовательностью бит, известной как преамбула, которая состоит из чередования единиц и нулей 101010... Войдя в битовую синхронизацию, приемник исследует входной поток на побитовой основе, пока не обнаружит байт начала кадра 10101011, который выполняет роль символа STX. За этим байтом следует заголовок кадра, в котором в определенном месте находится поле длины поля данных. Таким образом, в этой схеме приемник просто отсчитывает заданное количество байт, чтобы определить окончание кадра.

Третья схема (см. рис. 2.21, *в)* для обозначения начала и конца кадра использует флаги, которые включают запрещенные для данного кода сигналы (code violations, V). Например, при манчестерском кодировании вместо обязательного изменения полярности сигнала в середине тактового интервала уровень сигнала остается неизменным и низким (запрещенный сигнал J) или неизменным и высоким (запрещенный сигнал К). Начало кадра отмечается последовательностью JK0JK000, а конец – последовательностью JK1JK111. Этот способ очень экономичен, так как не требует ни битстаффинга, ни поля длины. Недостатком этого способа является то, что он зависит от принятого метода физического кодирования. При использовании избыточных кодов роль сигналов J и K играют запрещенные символы, например, в коде 4B/5B этими символами являются коды 11000 и 10001.

Каждая из трех схем имеет свои преимущества и недостатки. Флаги позволяют отказаться от специального дополнительного поля, но требуют специальных мер: либо по разрешению размещения флага в поле данных за счет битстаффинга, либо по использованию в качестве флага запрещенных сигналов, что делает эту схему зависимой от способа кодирования.

Протоколы с переменным форматом кадра

Существует ряд протоколов, в которых кадры имеют гибкую структуру. К таким протоколам относятся популярный прикладной протокол управления сетями SNMP и протокол канального уровня PPP, используемый для соединений типа «точка-точка». Кадры таких протоколов состоят из неопределенного количества полей, каждое из них может иметь переменную длину.

Для большей части протоколов характерны кадры, состоящие из служебных полей фиксированной длины. Исключение делается только для поля данных, так как возможны пересылки как небольших квитанций, так и больших файлов. Способ определения окончания кадра путем задания длины поля данных, рассмотренный выше, как раз рассчитан на такие кадры с фиксированной структурой и фиксированными размерами служебных полей.

Передача с установлением логического соединения и без установления логического соединения. При передаче кадров данных на канальном уровне используют дейтаграммные процедуры, работающие без установления логического соединения (connectionless), и процедуры с предварительным установлением логического соединения (connection-oriented) (рис. 2.22).

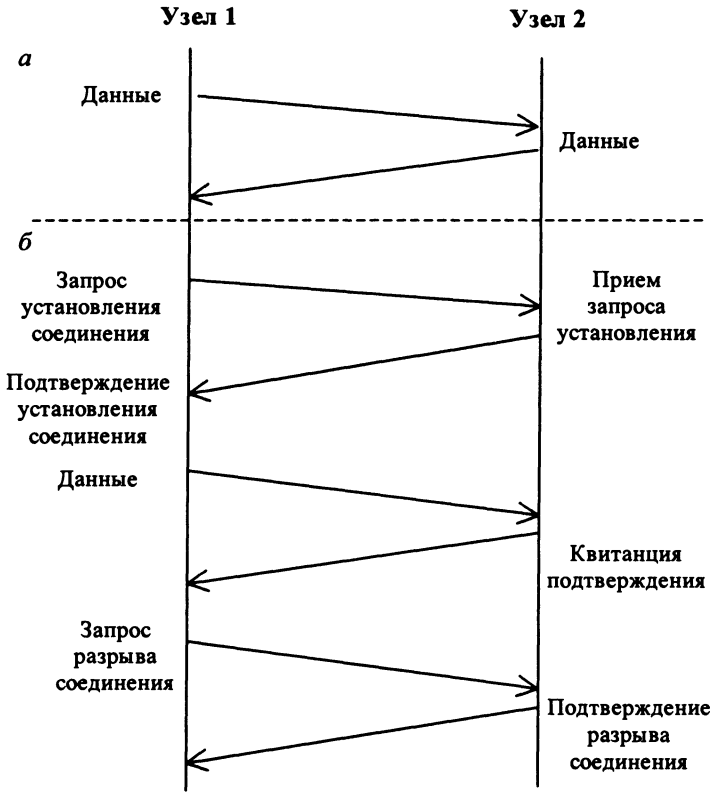


Рис. 2.22. Протоколы без установления соединения (*а*) и с установлением соединения (*б*)

При дейтаграммной передаче кадр посылается в сеть «без предупреждения», и никакой ответственности за его утерю протокол не несет (см. рис. 2.22, *а*). Предполагается, что сеть всегда готова принять кадр от конечного узла. Дейтаграммный метод работает быстро, так как никаких предварительных действий перед отправкой данных не выполняется. Однако при таком методе трудно организовать в рамках протокола отслеживание факта доставки кадра узлу назначения. Этот метод не гарантирует доставку пакета.

Передача с установлением соединения более надежна, но требует больше времени для передачи данных и вычислительных затрат от конечных узлов. При такой передаче узлу-получателю отправляется служебный кадр специального формата с предложением установить соединение (рис. 2.22, *б*). Если узел-получатель согласен с этим, то он посылает в ответ другой служебный кадр, подтверждающий установление соединения и предлагающий для данного логического соединения некоторые параметры, например идентификатор соединения, максимальное значение поля данных кадров, которые будут использоваться в рамках данного соединения, и т. п. Узел-инициатор соединения может завершить процесс установления соединения отправкой третьего служебного кадра, в котором сообщит, что предложенные параметры ему подходят. На этом логическое соединение считается установленным, и в его рамках можно передавать информационные кадры с пользовательскими данными. После передачи некоторого законченного набора данных, например определенного файла, узел инициирует разрыв данного логического соединения, посылая соответствующий служебный кадр.

В отличие от протоколов дейтаграммного типа, которые поддерживают только один тип кадра – информационный, протоколы, работающие по процедуре с установлением соединения, должны поддерживать несколько типов кадров – служебные, для установления (и разрыва) логического соединения, и информационные, переносящие собственно пользовательские данные.

Процедура установления соединения используется:

- для взаимной аутентификации либо пользователей, либо оборудования (маршрутизаторы тоже могут иметь имена и пароли, которые нужны для уверенности в том, что злоумышленник не подменил корпоративный маршрутизатор и не отвел поток данных в свою сеть для анализа);
- для согласования изменяемых параметров протокола: MTU, различных тайм-аутов и т. п.;
- для обнаружения и коррекции ошибок. Установление логического соединения дает точку отсчета для задания начальных значений номеров кадров. При потере нумерованного кадра приемник, во-первых, получает возможность обнаружить этот факт, а во-вторых, может сообщить передатчику, какой в точности кадр нужно передать повторно.

В некоторых технологиях процедуру установления логического соединения используют при динамической настройке коммутаторов сети для маршрутизации всех последующих кадров, которые будут проходить через сеть в рамках данного логического соединения. Так работают сети технологий X.25, Frame relay и ATM.

Обнаружение и коррекция ошибок

Канальный уровень должен обнаруживать ошибки передачи данных, связанные с искажением бит в принятом кадре данных или с потерей кадра, и по возможности их корректировать. Большая часть протоколов канального уровня выполняет только одну задачу – обнаружение ошибок, считая, что корректировать ошибки, т. е. повторно передавать данные, содержавшие искаженную информацию, должны протоколы верхних уровней. Так работают известные протоколы локальных сетей Ethernet, Token Ring, FDDI и др. Однако существуют протоколы канального уровня, например LLC2 или LAP-B, которые самостоятельно решают задачу восстановления искаженных или потерянных кадров.

Очевидно, что протоколы должны работать наиболее эффективно в типичных условиях работы сети. Поэтому для сетей, в которых искажения и потери кадров являются очень редкими событиями, разрабатываются протоколы типа Ethernet, где не предусмотрены процедуры устранения ошибок. Действительно, наличие процедур восстановления данных потребовало бы от конечных узлов дополнительных вычислительных затрат, которые в условиях надежной работы сети являлись бы избыточными.

Напротив, если в сети искажения и потери происходят часто, то желательно уже на канальном уровне использовать протокол с коррекцией ошибок, а не оставлять эту работу протоколам верхних уровней. Протоколы верхних уровней, например транспортного или прикладного, работая с большими тайм-аутами, восстановят утерянные данные с большой задержкой. В глобальных сетях первых поколений, например сетях X.25, которые использовали ненадежные каналы связи, протоколы канального уровня всегда выполняли процедуры восстановления потерянных и искаженных кадров. Поэтому нельзя считать, что один протокол лучше другого потому, что он восстанавливает ошибочные кадры, а другой протокол нет. Каждый протокол должен работать в тех условиях, для которых он разработан.

Методы коррекции ошибок, основанные на протоколах канального уровня, описаны в § 2.2. Рассмотрим вопрос восстанавления искаженных помехами данных.

Методы коррекции ошибок в вычислительных сетях основаны на повторной передаче кадра данных в случае, если кадр теряется и не доходит до адресата или приемник обнаружил в нем искажение информации. Чтобы убедиться в необходимости повторной передачи данных, отправитель нумерует отправляемые кадры и для каждого кадра ожидает от приемника так называемой *положительной квитанции* – служебного кадра, извещающего о том, что исходный кадр был получен и данные в нем оказались корректными. Время этого ожидания ограничено – при отправке каждого кадра передатчик запускает таймер, и, если по его истечении положительная квитанция не получена, кадр считается утерянным. Приемник в случае получения кадра с искаженными данными может отправить *отрицательную квитанцию* – что указывает на то,

что данный кадр нужно передать повторно. Процесс обмена квитанциями называется автоматическим запросом повторения – ARQ (Automatic Repeat reguest).

Существуют два подхода к организации процесса обмена квитанциями: с простоями и с организацией «скользящего окна».

Метод с простоями (Idle RQ) требует, чтобы источник, пославший кадр, ожидал получения квитанции (положительной или отрицательной) от приемника и только после этого посылал следующий кадр (или повторял искаженный). Если же квитанция не приходит в течение тайм-аута, то кадр (или квитанция) считается утерянным и его передача повторяется. В этом случае производительность обмена данными существенно снижается, так как передатчик не может послать следующий кадр сразу же после отправки предыдущего, он обязан ждать прихода квитанции. Снижение производительности этого метода коррекции особенно заметно на низкоскоростных каналах связи (в территориальных сетях).

В *методе «скользящего окна»* (sliding window) для повышения коэффициента использования линии источнику разрешается передать некоторое количество кадров в непрерывном режиме, т. е. в максимально возможном для источника темпе, без получения на эти кадры положительных ответных квитанций. Количество кадров, которые разрешается передавать таким образом, называется размером окна. Рис. 2.23, *а* иллюстрирует данный метод для передающего окна размером в W кадров. В начальный момент, когда еще не послано ни одного кадра, окно определяет диапазон кадров с номерами от 1 до W включительно. Источник начинает передавать кадры и получать в ответ квитанции. Квитанции канального уровня поступают в той же последовательности, что и кадры, которым они соответствуют. В момент t_1 при получении первой квитанции K_1 окно сдвигается на одну позицию, определяя новый диапазон от 2 до $(W + 1)$.

Процессы отправки кадров и получения квитанций происходят независимо друг от друга. Рассмотрим произвольный момент времени t_n, когда источник получил квитанцию на кадр с номером n. Окно сдвинулось вправо и определило диапазон разрешенных к передаче кадров от $(n + 1)$ до $(W + n)$. Все множество кадров, выходящих из источника, можно разделить на перечисленные ниже группы (см. рис. 2.23, *а*).

Кадры с номерами от 1 до n уже были отправлены и квитанции на них получены, т. е. они находятся за пределами окна слева.

Кадры, начиная с номера $(n + 1)$ и кончая номером $(W + n)$, расположены в пределах окна и потому могут быть отправлены, не дожидаясь прихода какой-либо квитанции. Этот диапазон можно разделить еще на два поддиапазона:

кадры с номерами от $(n + 1)$ до m, которые уже отправлены, но квитанции на них еще не получены;

кадры с номерами от m до $(W + n)$, которые пока не отправлены, хотя запрета на это нет.

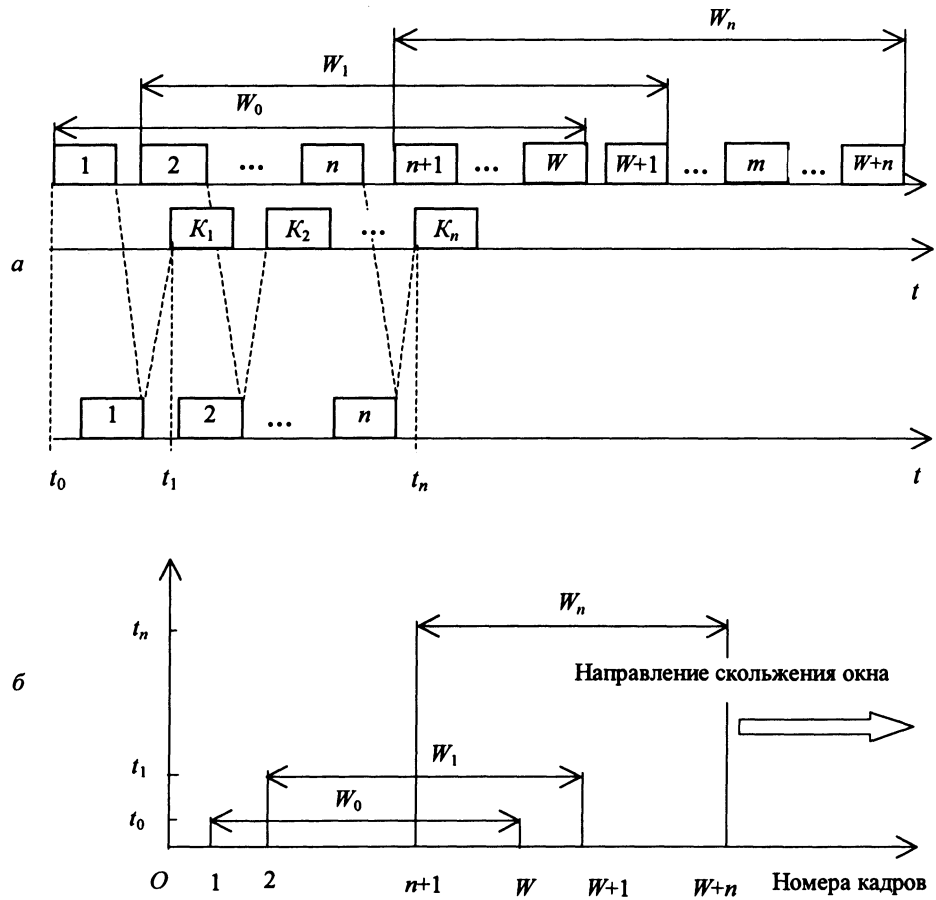


Рис. 2.23. Метод скользящего окна:
t_0 – исходный момент, t_1 и t_n – моменты прихода квитанций на 1- и n-й кадр соответственно

Все кадры с номерами, большими или равными ($W + n + 1$), находятся за пределами окна справа и поэтому пока не могут быть отправлены.

Перемещение окна вдоль последовательности номеров кадров показано на рис. 2.23, *б*. Каждый раз, когда приходит положительная квитанция, окно сдвигается, но его размер при этом не меняется и остается равным W. При отправке кадра с номером n источнику разрешается передать еще $W - 1$ кадров до получения квитанции на кадр n, так что в сеть последним уйдет кадр с номером ($W + n - 1$). Если же за это время квитанция на кадр n так и не пришла, то процесс передачи приостанавливается, и по истечении некоторого тайм-аута кадр n (или квитанция на него) считается утерянным, и его передают снова. Если поток подтверждений поступает регулярно, в пределах допуска в W кадров, то скорость обмена достигает максимально возможной для данного канала и принятого протокола.

Метод «скользящего окна» более сложен в реализации, чем метод с простоями, так как передатчик должен хранить в буфере все кадры, на которые пока не получены положительные квитанции. Кроме того, требуется отслеживать несколько параметров алгоритма: размер окна W, номер кадра, на который получена квитанция, номер кадра, который еще можно передать до получения новой квитанции. Приемник может не посылать квитанции на каждый принятый корректный кадр. Если несколько кадров пришли почти одновременно, то приемник может послать квитанцию только на последний кадр. При этом подразумевается, что все предыдущие кадры также дошли благополучно.

Некоторые методы используют отрицательные квитанции, которые бывают двух типов – групповые и избирательные. Групповая квитанция содержит номер кадра, начиная с которого нужно повторить передачу всех кадров, отправленных передатчиком в сеть. Избирательная квитанция требует повторной передачи только одного кадра.

Метод с простоями является частным случаем метода «скользящего окна», когда размер окна равен единице. Метод «скользящего окна» имеет два параметра, которые могут заметно влиять на эффективность передачи данных между передатчиком и приемником, – размер окна и величина тайм-аута ожидания квитанции. В надежных сетях, где кадры искажаются и теряются редко, для повышения скорости обмена данными размер окна можно увеличивать, так как при этом передатчик будет посылать кадры с меньшими паузами. В ненадежных сетях размер окна следует уменьшать, так как при частых потерях и искажениях кадров резко возрастает объем вторично передаваемых через сеть кадров, а значит, пропускная способность сети будет расходоваться во многом вхолостую – полезная пропускная способность сети будет падать.

Выбор тайм-аута зависит не от надежности сети, а от задержек передачи кадров сетью. Во многих реализациях метода «скользящего окна» величина окна и тайм-аут выбираются адаптивно, в зависимости от текущего состояния сети.

Телефонный канал для передачи данных

Развитие информационных сетей предполагает в качестве непременного условия наличие высоконадежных скоростных систем передачи данных (ПД). В отечественной практике ПД вне локальных сетей на расстояния, превышающие 1000 м, обычно проводится с использованием традиционной коммутируемой телефонной сети или посредством подключения абонентов к выделяемым магистральным телефонным каналам. На стыке цифрового потока данных и аналогового телефонного канала устанавливают модем, что вызывает проблему обеспечения максимально возможной скорости и надежности ПД.

Российские магистральные системы связи более чем на 90 % состоят из аналоговых систем передачи, которые существенным образом искажают форму передаваемого сигнала. В местных системах связи в ряде случаев используют устаревшее коммутационное оборудование, что приводит к существенному ослаблению сигнала и внесению в сигнал мощных помех.

Искажения сигнала в телефонном канале. Сигнал, передаваемый по телефонному каналу, по пути следования к точке приема подвергается не только воздействию шумов, но, как правило, претерпевает и более сложные искажения.

Выделяют два основных требования к параметрам модемов:

- порог чувствительности модемов с запасом 5...10 дБ должен превышать значение амплитудно-частотной характеристики (АЧХ) на краях диапазона частот;
- уровень мощности передаваемого в линию сигнала должен иметь значение, при котором еще обеспечивается линейность линии, т. е. разность уровней основной и высших гармоник сигнала в точке приема не менее 40 дБ,

а также факторы, влияющие на устойчивость работы модемов по каналам тональной частоты:

- наличие импульсных помех на уровне 3...10 дБ (в зависимости от порога устойчивости используемых модемов) выше уровня принимаемого сигнала является одной из причин частых «зависаний» модема;
- перерывы связи длительностью около 1 мс (уровень фиксации ниже принимаемого сигнала на 3...30 дБ на частоте 1020 Гц) объясняют факты срывов сеансов;
- модемы плохо переносят скачки фазы и амплитуды, превышающие пороговые значения 10...20 угловых градусов и 2...6 дБ соответственно. В лучшем случае последствием таких скачков может быть пересогласование параметров (*retrain*) с успешным завершением, в худшем – модем «виснет».

При использовании протяженных линий связи, образованных четырехпроводными магистральными каналами с двухпроводными окончаниями в точках подключения оборудования пользователя возникают эхо-отражения передаваемого и принимаемого сигнала (рис. 2.24).

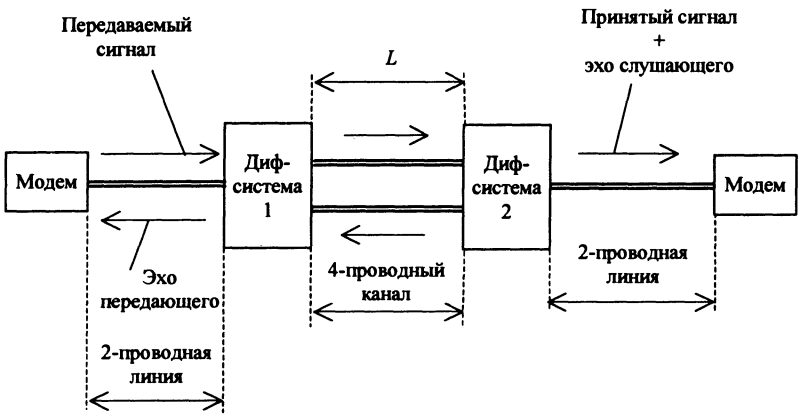


Рис. 2.24. Возникновение эхо-сигналов

Эхо-сигнал передающего образуется при отражении сигнала передающей стороны от удаленной системы разделения направлений передачи (дифсистема 2). Задержка T, мс, прямо пропорциональна удвоенной длине канала L, км и обратно пропорциональна скорости распространения сигнала C, км/мс:

$$T = 2L/C. \tag{3.96}$$

Эта формула не учитывает задержек сигнала в каналообразующей аппаратуре. Эхо-сигнал слушающего образуется при двойном отражении – сигнал передающей стороны отражается от удаленной разделительной системы (дифсистема 2), возвращается обратно, отражается от ближней разделительной системы (дифсистема 1) и, поступая на вход принимающей стороны, складывается с переданным сигналом. Задержка эхо-сигнала слушающего относительного основного переданного сигнала определяется той же формулой. Так при использовании спутниковых каналов (при высоте геостационарной орбиты спутника L = 36000 км) задержки эхо передающего (говорящего) и принимающего (слушающего) составят:

$$T_г = T_с = 2 \cdot 36000/300 = 240 \text{ мс}. \tag{3.97}$$

Места возникновения эха условно подразделяются на «ближнее» эхо (влияние собственной дифсистемы) и «дальнее» эхо (воздействие удаленной дифсистемы). Методы построения эхо-конпенсаторов (ЭК) «ближнего» и «дальнего» эха существенно отличаются, так как «ближнее» эхо имеет малое время задержки (относительно передаваемого сигнала) и высокий уровень воздействия (уровень эхо-сигнала (ЭС) может превышать принимаемый сигнал на 30...40 дБ).

При передаче данных с использованием модемов с эхо компенсацией (V.32, V.32bis, V.34) необходимо использовать линии с задержкой эхо передающего не более 300 мс, причем количество эхо-сигналов должно быть не более одного. При использовании модемов с частотным разделением каналов приема-передачи (V.22, V.22bis) эхо передающего модема не оказывает дестабилизирующего воздействия на приемник. Эхо-сигнал принимающего оказывает негативное воздействие на приемник при любом способе модуляции, независимо от времени задержки. Для уменьшения этого воздействия необходимо, чтобы мощность эхо-сигнала принимающего была бы ниже уровня мощности основного сигнала более чем на 15 дБ для скоростей 1200 бит/с (V.22) и 4800 бит/с (V.32) и более чем на 25...30 дБ для скоростей 2400 бит/с (V.22bis), 9600 бит/с и высших (V.32, V.32bis, V.34).

Технологии передачи данных по телефонным каналам связи

Стандартные протоколы модемной связи. *Протокол V.21* – дуплексный протокол с частотным разделением каналов и частотной модуляцией FSK. На нижнем канале (его обычно использует для передачи вызывающий модем) 1 передается частотой 980 Гц, а 0 – 1180 Гц. На верхнем канале (передает отвечающий) 1 передается частотой 1650 Гц, а 0 – 1850 Гц. Модуляционная и

информационная скорости соответственно равны 300 бод и 300 бит/с. Несмотря на невысокую скорость, данный протокол находит применение прежде всего в качестве «аварийного», при невозможности из-за высокого уровня помех использовать другие протоколы физического уровня. Кроме того, ввиду своей неприхотливости и помехоустойчивости, он используется в специальных высокоуровневых приложениях, требующих высокой надежности передачи. Например, при установке соединения между модемами по Рекомендации *V.8*, или для передачи управляющих команд при факсимильной связи (верхний канал).

Протокол V.22 – дуплексный протокол с частотным разделением каналов и модуляцией DPSK. Несущая частота нижнего канала (передает вызывающий) – 1200 Гц, верхнего (передает отвечающий) – 2400 Гц; модуляционная скорость – 600 бод. Имеет режимы двухпозиционной (кодируется бит) и четырехпозиционной (дибит) фазоразностной модуляции с фазовым расстоянием между точками, соответственно, 180 и 90 град. Соответственно, информационная скорость может быть 600 или 1200 бит/с. Этот протокол фактически поглощен протоколом V.22bis.

Протокол V.22bis – дуплексный протокол с частотным разделением каналов и модуляцией QAM. Несущая частота нижнего канала (передает вызывающий) составляет 1200 Гц, верхнего – 2400 Гц; модуляционная скорость – 600 бод. Имеет режимы 4-позиционной (кодируется дибит) и 16-позиционной (кодируется квадробит) квадратурной амплитудной модуляции. Соответственно, информационная скорость равна 1200 или 2400 бит/с. Режим 1200 бит/с полностью совместим с V.22, несмотря на другой тип модуляции, так как первые два бита в режиме 16-QAM (квадробит) определяют изменение фазового квадранта относительно предыдущего сигнального элемента и потому за амплитуду не отвечают, а последние два бита определяют положение сигнального элемента внутри квадранта с вариацией амплитуды. Таким образом, *DPSK* можно рассматривать как частный случай QAM, где два последних бита не меняют своих значений, в результате, из 16 позиций выбираются четыре в разных квадрантах, но с одинаковым положением внутри квадранта, в том числе и с одинаковой амплитудой. Протокол V.22bis является стандартом де-факто для всех среднескоростных модемов.

Протокол V.32 – дуплексный протокол с эхо-компенсацией и квадратурной амплитудной модуляцией или модуляцией с решетчатым кодированием. Частота несущего сигнала равна 1800 Гц, модуляционная скорость – 2400 бод. Имеет режимы двухпозиционной (бит), четырехпозиционной (дибит) и 16-позиционной (квадробит) QAM. Соответственно, информационная скорость может быть 2400, 4800 и 9600 бит/с. Кроме того, для скорости 9600 бит/с имеет место альтернативная модуляция – 32-позиционная с применением треллис-кодирования (32-TCM). Полоса частот, занимаемая сигналом, составляет от 600 до 3000 Гц. Реализация сигнально-кодовой конструкции 32-TCM связана с внесением одного избыточного бита в расчете на один сигнальный отсчет. В результате этого каждый сигнальный отсчет 5 бит информации. Скорость передачи

остается равной 9600 бит/с за счет того, что число возможных сигнальных позиций увеличено ровно в 2 раза и стало равным 32. Такой режим работы позволяет значительно повысить помехоустойчивость передачи.

Протокол V.32bis. Протокол модуляции V.32bis разработан для обеспечения передачи данных со скоростью до 14400 бит/с по двухпроводным коммутируемым и выделенным телефонным каналам. Это дуплексный протокол с эхокомпенсацией и модуляцией ТСМ. Используются те же, что в V.32, частота несущего сигнала – 1800 Гц, модуляционная скорость – 2400 бод и полоса частот сигнала – 600...3000 Гц. Имеет режимы 16-ТСМ, 32-ТСМ, 64-ТСМ и 128-ТСМ. Соответственно, информационная скорость может быть 7200, 9600, 12000 и 14400 бит/с. Скорость передачи без треллис-кодирования – 4800 бит/с. Совместим с *V.32* на скоростях 4800 и 9600 бит/с. Режим асимметрической передачи не поддерживается, т. е. скорость передачи и приема каждого взаимодействующего модема должны быть одинаковы. Согласно протоколу *V.32bis* модемы должны иметь два самосинхронизирующихся скрэмблера. В каждом направлении передачи используется свой скрэмблер. Вызывающий модем использует скрэмблер с образующим полиномом $+ x^{-18} + x^{-23}$, а отвечающий – скремблер с образующим полиномом $+ x^{-5} + x^{-23}$.

Протокол V.34. Рекомендация V.34 регламентирует процедуры передачи данных по коммутируемым телефонным каналам со скоростями до 28800 бит/с. Скорость передачи данных выбирается из множества допустимых значений в диапазоне от 2400 до 28800 бит/с с шагом 2400 бит/с. Таким образом возможен выбор 12 значений, а также изменение скорости передач в процессе сеанса связи. В отличие от более ранних протоколов, скорость модуляции не является фиксированной величиной. Рекомендация предусматривает шесть скоростей модуляции, равных 2400, 2743, 2800, 3000, 3200 и 3429 символов в секунду. В Рекомендации V.34 вместо единицы измерения скорости модуляции «бод» введено понятие «символ в секунду».

Для повышения скорости передачи необходимо выбирать большее значение скорости модуляции. Однако для полосы пропускания стандартного телефонного канала 3100 Гц две последние модуляционные скорости являются неприемлемыми. Тем не менее, работа на таких скоростях возможна благодаря не идеальности характеристик фильтров каналообразующей аппаратуры.

При введении таких «запредельных» скоростей была учтена тенденция увеличения в коммутируемой телефонной сети общего пользования (КТСОП) доли систем передачи с ИКМ, в которых реальная полоса пропускания телефонного канала может достигать 3500 Гц. Кроме того, при установлении соединения через КТСОП в пределах города канал связи чаще всего представляет собой соединение нескольких физических (кабельных) линий. Такой канал при наличии специальных средств частотной коррекции также может обеспечить передачу сигнала с более широким спектром. Для канала, не позволяющего расширить стандартную полосу пропускания, максимально допустимой символьной скоростью является значение 3000 символов в 1 с.

В отличие от протокола V.32 , в V.34 увеличена размерность кодируемого информационного элемента многопозиционной QAM с треллис-кодированием. В предыдущих протоколах QAM информационный элемент был двумерным, так как значение элемента характеризовалось амплитудой и фазой сигнала. Рекомендация V.34 предусматривает для описания информационного элемента использование третьего параметра – времени, который порождает еще два измерения информационного элемента. Таким образом, в четырехмерном пространстве каждый информационный элемент (сигнальная точка) имеет четыре координаты и передается за два символьных интервала. В этом случае каждый кодируемый элемент включает в себя два последовательно передаваемых символа, представляющих собой сигналы, промодулированные по амплитуде и фазе. В самой Рекомендации V.34 представлено 50 различных сигнальных диаграмм (*созвездий*), которые обеспечивают работу на всех скоростях. Переход к четырехмерным сигнальным кодовым конструкциям позволил существенно увеличить общее число сигнальных точек, что, в свою очередь, повысило скорость без ухудшения помехоустойчивости. При формировании позиционного номера сигнальной точки, как и ранее, применяют лишь один избыточный бит решетчатого кодера.

В V.34 сделан шаг вперед и в области треллис-кодирования. Здесь использован сверточный код на 16, 32, 64 состояния, что позволяет повысить помехоустойчивость всей системы сигналов за счет увеличения свободного евклидового пространства между соседними путями на решетчатой диаграмме. Однако это приводит к увеличению задержки на принятие решения и к повышению требований к объему памяти и вычислительной мощности процессора модема.

Значение частоты несущей согласно V.34 не фиксированно, оно выбирается из ряда: 1600, 1646, 1680, 1800, 1829, 1867, 1920, 1959, 2000 Гц. Большое число возможных значений скорости модуляции, скорости передачи и несущей частоты представляет модему возможность использовать имеющуюся полосу частот с максимальной эффективностью.

Нововведение протокола V.34 в области организации дуплексной связи заключается в его асимметричности по многим параметрам. Передача данных между двумя модемами V.34 может осуществляться не только с разными скоростями, но и на разных частотах с использованием различных сигнальных диаграмм. В стандарте также предусмотрен режим полудуплексной передачи, которая предполагает взаимодействие модемов без схем эхокомпенсации. В предыдущих поколениях протоколов адаптивная подстройка под конкретные характеристики канала осуществлялась исключительно на приемной стороне. В V.34 идея адаптации носит глобальный характер.

При QAM с большим сигнальным пространством диапазон возможных амплитуд сигналов довольно велик. Из-за этого может возникнуть статистическая зависимость между передаваемой информацией и уровнем сигнала на выходе. Что может повлечь за собой ситуации, при которых выходной сигнал будет

иметь малую амплитуду в течение длительного времени. В таких ситуациях возможны сбои декодера и потеря сигнала на приемной стороне. Также возможно формирование сигнала с большим пик-фактором (отношение пикового значения мощности к среднему значению), что приводит к ухудшению общих характеристик системы (увеличивает уровень взаимных и нелинейных искажений). Для решения этой проблемы Рекомендация V.34 предлагает специальное предкодирование, в котором двумерное созвездие разбивается на концентрические кольца, содержащие равные количества сигнальных точек с близкой или одинаковой амплитудой. В передающую часть модема V.34 введен генератор колец, способствующий синтезу требуемой формы выходного сигнала.

Протокол V.34 обеспечивает амплитудно-фазовую предкоррекцию сигнала передатчика для устранения межсимвольной интерференции. Эта предкоррекция позволяет получить выигрыш более 3,5 дБ по сравнению с линейной коррекцией, применяемой в протоколе V.32. Предварительное искажение на передающей стороне вводится с помощью специального цифрового фильтра, значения коэффициентов фильтрации передаются от удаленного модема на этапе вхождения в связь. В результате этой процедуры передаваемый сигнал приобретает искажения, компенсирующие те, которые он получил при прохождении по каналу. Это существенно облегчает работу адаптивного эквалайзера на приемной стороне. Дополнительно в протокол заложена возможность выбора одного из 11 заранее заданных шаблонов для спектра передатчика. Такие шаблоны предусматривают подъем высокочастотных составляющих спектра, что компенсирует искажения, вносимые абонентскими и соединительными линиями.

В протоколе V.34 впервые используется иерархическая кадровая структура на физическом уровне. Сигнальные кадры, состоящие из 4 четырехмерных информационных элементов (8 символов), объединяются в кадры данных, которые, в свою очередь, составляют суперкадр. В систему введены средства для поддержания синхронизации по кадрам благодаря чему суперкадр имеет фиксированную длительность 280 мс.

Широкие возможности адаптации предусмотрены и на этапе вхождения в связь. Процедура вхождения в связь состоит их четырех фаз. На первой фазе модемы выбирают наивысший протокол серии V, реализованный в обоих модемах. На этом этапе соединение устанавливается согласно Рекомендациям V.25 и V.8 . Если оба модема поддерживают протокол V.34, то они переходят ко второй фазе, в ходе которой классифицируется канал связи. На 3- и 4-й фазах происходит обучение адаптивного эквалайзера, эхокомпенсатора и других систем модема.

После установления соединения осуществляется процедура адаптации к каналу связи – передатчик модема посылает в линию специальный тестовый сигнал, представляющий собой последовательность из 21 гармонического колебания разных частот в диапазоне от 150 до 3750 Гц. Приемник удаленного модема, принимая этот сигнал, рассчитывает частотную характеристику ка-

нала связи, степень нелинейных искажений, сдвиг частот и ряд других характеристик канала. Затем выбирается номинальная скорость модуляции, значение несущей частоты, уровень передачи, номер шаблона и коэффициенты предкорректора, скорость передачи данных, число состояний решетчатого кодера, тип сигнально-кодовой конструкции и другая информация о желаемой конфигурации удаленного передатчика. Такая же процедура выполняется и в противоположном направлении. Затем оба модема обмениваются этими установками. Для этого используют протоколы V.22 (600 бит/с) и V.42.

Рекомендация V.34 реализует системный подход к решению проблемы помехоустойчивости. Поэтому модем V.34 может работать с большей скоростью, чем другие на каналах такого же качества. В 1996 г. введена поправка к стандарту V.34, предусматривающая возможность передачи данных со скоростью 33,6 кбит/с. Модемы, поддерживающие такую скорость, часто называют модемами V.34+ или V.34bis.

Протокол V.23. Это полудуплексный протокол с частотной модуляцией FSK. В нем имеется два скоростных режима: 600 и 1200 бит/с. Модуляционная и информационная скорости соответственно равны 600 и 1200 бод. В обоих режимах 1 передается частотой 1300 Гц. В режиме 600 бит/с 0 передается частотой 1700 Гц, а в режиме 1200 бит/с – частотой 2100 Гц. Реализация протокола опционально может включать обратный канал, работающий на скорости 75 бит/с, что превращает протокол в асимметричный дуплексный. Частота передачи 1 в обратном канале – 390 Гц, 0 – 450 Гц. Этот протокол практически не употребляют в качестве стандартного протокола модемной связи, и далеко не всякий стандартный модем им оснащен. Однако он служил и до сих пор остается базовым для реализации нестандартных модемов, получивших широкое распространение в нашей стране (типа *LEXAND*). Видимо, благодаря простоте, высокой помехоустойчивости и приличной (по сравнению с V.21) скорости. Кроме того, в ряде европейских стран этот протокол используют в информационной системе Videotex.

Протоколы V.26, V.26bis, V.26ter. Эти три протокола объединяет тип модуляции (DPSK), частота несущей (1800 Гц) и модуляционная скорость (1200 бод). Разница между ними заключается в возможности и способах обеспечения дуплексной связи и в информационной скорости. Протокол V.26 обеспечивает дуплекс только по четырехпроводной выделенной линии. Протокол V.26bis – это полудуплексный протокол, предназначенный для работы по двухпроводной коммутируемой линии, а V.26ter обеспечивает полный дуплекс с помощью технологии эхо-компенсации. Кроме того, первые два протокола могут быть асимметричными дуплексными, опционально включая обратный канал, работающий на скорости 75 бит/с в соответствии с V.23. Все три протокола обеспечивают скорость передачи информации 2400 бит/с посредством четырехпозиционной (дибит) DPSK. V.26bis и V.26ter, кроме того, они имеют режим двухпозиционной (бит) DPSK, обеспечивая скорость 1200 бит/с.

Протокол V.33. В этом протоколе используется модуляция с решетчатым кодированием TCM. Он предназначен для обеспечения дуплексной связи на четырехпроводных выделенных каналах, имеет частоту несущего сигнала 1800 Гц и модуляционную скорость 2400 бод. Работает в режимах 64-TCM и 128-TCM. Соответственно, информационная скорость равна 12000 и 14400 бит/с. Этот протокол очень напоминает V.32bis без эхо-компенсации. Более того, если модем с протоколом V.33 установить на четырехпроводное окончание до дифференциальной системы АТС, то он вполне сможет связаться с удаленным модемом V.32bis, установленным на двухпроводной линии.

Нестандартные протоколы. *ZyX.* Протокол разработан фирмой ZyXEL Communications Corporation и реализован в собственных модемах. Он также, как и V.32ter, расширяет V.32bis значениями информационных скоростей 16800 и 19200 бит/с с сохранением технологии эхо-компенсации, модуляции с трелис-кодированием и несущей 1800 Гц. Модуляционная скорость 2400 бод сохраняется лишь для 16800 бит/с. Скорость 19200 бит/с обеспечивается повышением модуляционной скорости до 2743 бод при сохранении режима модуляции 256-TCM для обеих скоростей. Такое решение позволяет снизить требование к отношению сигнал/шум на линии на 2,4 дБ, однако расширение полосы пропускания может негативно сказываться при больших искажениях амплитудно-частотной характеристики канала.

HST. Протокол HST (High Speed Technology) разработан фирмой U.S.Robotics и реализован в модемах фирмы серии Courier и World Port. Это – асимметричный дуплексный протокол с частотным разделением каналов. Обратный канал работает в режимах 300 и 450 бит/с; основной канал – 4800, 7200, 9600, 12000, 14400 и 16800 бит/с. Последняя версия этого протокола поддерживает скорости до 21600 бит/с. Протокол HST характеризуется сравнительной простотой, высокой помехоустойчивостью вследствие отсутствия необходимости в эхокомпенсации и отсутствия взаимовлияния каналов. Благодаря этому модемы с HST иногда показывают даже лучшие результаты, чем устройства с V.34.

Отличительной особенностью HST, обусловливающей его высокую устойчивость является несимметричность. При передаче данных скорости передатчика и приемника не совпадают. Модем передает данные на скорости 16800 бит/с в одну сторону и на скорости 450 бит/с – в другую, автоматически переключая направление скоростного канала в зависимости от количества передаваемых данных. Все скоростные двунаправленные протоколы передачи данных (V.32, V.32bis и др.) при значительных сбоях в линии вынуждены затрачивать длительное время (около 6...8 с) на подстройку параметров (в основном параметров эхокомпенсации), необходимой для двунаправленной передачи данных (эта операция называется ретрейн – retrain). Протокол HST требует значительно меньше времени для выполнения этой операции – порядка 0,5 с, благодаря чему существенно повышается скорость и надежность работы при частых помехах в линии. К сожалению, на линиях со слабыми, но постоянными

помехами, когда из-за ошибок модем вынужден постоянно передавать большие объемы данных повторно, скорость передачи данных может оказаться ниже, чем на двунаправленных протоколах (без снижения надежности работы). Это вызвано тем, что из-за низкой пропускной способности обратного канала на HST, данные передаются большими по сравнению с V.32 блоками (это снижает объем служебной информации, передаваемой по низкоскоростному каналу, но увеличивает объем повторно передаваемых данных). На таких линиях может оказаться целесообразным использовать двунаправленные протоколы вместо HST. При работе с некоторыми коммуникационными пакетами, передающими файлы одновременно в две стороны, несимметричность HST может вызвать сбои в работе программы (из-за очень медленной передачи данных в одном направлении), кроме того такой режим работы вызывает частые неоправданных переключения направления каналов, что сильно снижает скорость.

PEP, TurboPEP. Полудуплексные протоколы семейства PEP (Packetized Ensemble Protocol) разработаны фирмой Telebit и реализованы в модемах фирмы серий TrailBlazer (PEP) и WorldBlazer (TurboPEP). В этих протоколах принципиально иным образом используется вся полоса пропускания канала тональной частоты для высокоскоростной передачи данных. Весь канал разбивается на множество узкополосных частотных подканалов, по каждому из которых независимо передается своя порция бит из общего потока информации. Такого рода протоколы называют многоканальными, или параллельными, или протоколами с множеством несущих (multicarrier). В протоколе PEP канал разбивается на 511 подканалов. В каждом подканале шириной около 6 Гц с модуляционной скоростью от 2 до 6 бод с помощью квадратурной амплитудной модуляции кодируются от 2 до 6 бит на бод. Предусмотрено несколько степеней свободы для обеспечения максимальной пропускной способности каждого конкретного канала, имеющего свои характеристики. В процессе установки соединения каждый частотный подканал независимо тестируется и определяется возможность его использования, а также параметры: модуляционная скорость подканала и число позиций модуляции. Максимальная скорость передачи по протоколу PEP составляет 19200 бит/с. В процессе сеанса при ухудшении качества телефонного канала параметры подканалов могут меняться, а некоторые подканалы отключаться. При этом декремент понижения скорости не превышает 100 бит/с. Протокол TurboPEP за счет увеличения числа подканалов, а также количества кодируемых на одном бодовом интервале бит, может достигать скорости 23000 бит/с. Кроме того, в протоколе TurboPEP применяется модуляция с треллис-кодированием, что увеличивает помехоустойчивость протокола.

Основными преимуществами протоколов PEP и TurboPEP является слабая чувствительность к искажениям АЧХ канала и значительно меньшая чувствительность к импульсным помехам по сравнению с традиционными протоколами. Если первое не вызывает вопросов, то в части импульсных помех необходимы некоторые комментарии. Дело в том, что хотя импульсная помеха практически перекрывает всю ширину спектра, т. е. по всем подканалам, но в

связи со значительно большей длительностью сигнала по сравнению с традиционными протоколами (6 бод против 2400), искаженная помехой доля сигнала много меньше, что разрешает в ряде случаев безошибочно его демодулировать.

Многоканальные протоколы позволяют успешно работать даже на линиях, где установлены режекторные фильтры, ограничивающие использование телефонных каналов для передачи данных с помощью стандартных модемов.

K56-технология (56Kflex, x2, V.90). Модемы всех стандартов, включая V.34, разрабатывались с учетом того, что в КТСОП установлено как аналоговое, так и цифровое оборудование, соединенное между собой цифровыми и аналоговыми линиями. Таким образом, во время путешествия от передающего к приемному модему сигнал подвергается многочисленным преобразованиям из аналоговой формы в цифровую и обратно, а также влиянию различных помех на аналоговых участках пути. Все эти преобразования не позволяют обеспечить величину отношения полезного сигнала к шумам выше 30...35 дБ. В соответствии с законом Шеннона-Хартли теоретически возможная скорость по стандартному аналоговому или смешанному телефонному каналу с таким отношением сигнал/шум будет не выше 35 кбит/с.

В 1998 г. ITU-T принял стандарт V.90, который является компромиссным решением двух конкурирующих технологий модемного соединения на скорости 56 кбит/с – технологии *x*2 фирм US Robotics и 3Com, и технологии 56Kflex фирм Lucent и Rockwell. Протокол V.90 стал единым стандартом для всех производителей и включает в себя лучшие технические решения обоих конкурирующих протоколов.

K56-технология служит своеобразным мостом между современными КТСОП и полностью цифровыми сетями, например ISDN. Она обеспечивает увеличение скорости получения данных без дополнительных капиталовложений на организацию цифровых абонентских линий. С ее помощью пользователи Internet могут значительно быстрее загружать на свой компьютер графические Web-страницы, аудио- и видеофайлы, т. е. данные, для транспортировки которых в случае применения модемов стандарта V.34 требуется продолжительное время.

Технология передачи информации на скорости 56 кбит/с несколько отличается от технологии, применяемой в модемах со скоростями 33,6 кбит/с и ниже. При традиционном способе передачи информация, представленная в компьютере в цифровом виде, с помощью модема преобразуется в аналоговый сигнал, который проходит через аналоговую телефонную линию на телефонную станцию. На телефонной станции аналоговый сигнал преобразуется в цифровую форму и передается по оптоволоконному каналу в уплотненном виде на другую станцию, где он разворачивается и вновь преобразуется в аналоговую форму. Затем по аналоговой линии сигнал абонента передается к другому модему, преобразующему полученный сигнал в цифровую форму и передающему полученную информацию в компьютер. В итоге получается, что данные на пути к месту назначения проходят два цифро-аналоговых и два аналого-цифровых преобразования (рис. 2.25).

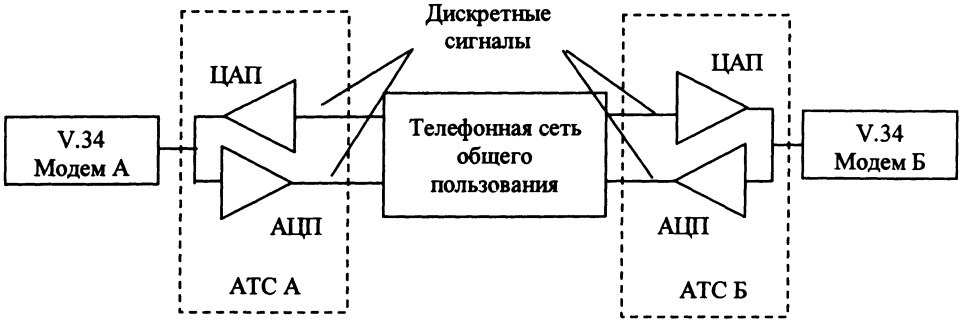


Рис. 2.25. Подключение модема к аналоговой сети

Аналоговая информация из модема преобразуется в цифровую форму, чтобы ее можно было передать через цифровые каналы городской телефонной сети. Входящий в АЦП аналоговый поток квантуется с частотой 8000 раз в 1 с и каждый раз амплитуда сигнала записывается как ИКМ-код. Цифровой поток, отправленный через городскую телефонную сеть, восстанавливается на другом конце приблизительно в виде исходного аналогового сигнала. Разница между оригинальным и восстановленным сигналом называется шумом квантования и ограничивает скорость модема примерно до 35 кбит/с. Но шум квантования имеет место только при аналого-цифровых преобразованиях и не сказывается на цифро-аналоговых. Это и является ключом к К56-технологии: если не будет аналого-цифровых преобразований между серверным модемом и городской телефонной сетью, то отправленный цифровой сигнал дойдет до модема на стороне клиента, без каких-либо потерь.

Внутри модемов преобразование из аналогового сигнала в цифровой происходит практически без появления шумов, так как в модемах применяют АЦП с большей разрядностью, чем на телефонной станции, и значения младших «шумящих» битов отбрасываются. При достижении определенного порога (называемого порогом Шеннона) соотношение сигнал/шум становится слишком низким для качественной передачи данных.

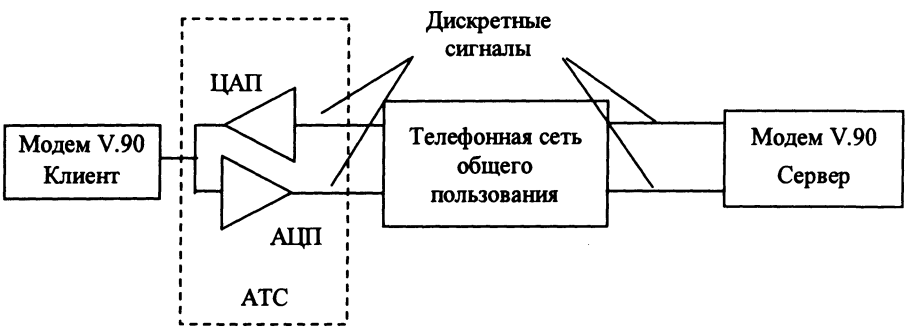


Рис. 2.26. Подключение модема V.90 к сети с одним АЦП

K56-технология предназначена для телефонных сетей общего пользования, в которых остался аналоговым только небольшой абонентский участок – от местной АТС до квартиры пользователя. Вся же транспортная сеть, оборудование АТС, узлов провайдеров Internet и крупных компаний, а также линии связи, соединяющие эти узлы с ближайшими АТС, должны быть полностью цифровыми (рис. 2.26). В этой технологии используют модемы V.90 двух типов: серверные и клиентские, которые в полном смысле слова не являются модемами. Для их обозначения используют термин «ИКМ-модем». Серверные модемы устанавливают у провайдеров, и они образуют модемный пул этих провайдеров. Клиентские модемы устанавливают на рабочих местах пользователей. При этом, если качество абонентской линии обеспечивает установление связи по скоростному протоколу (в частности, ее длина не должна превышать 3 км), то передача нисходящего трафика (в направлении пользователя) осуществляется следующим образом. Восьмизначные ИКМ-коды, используемые в цифровой части телефонной сети, передаются по цифровой сети до ближайшей к пользователю АТС. ЦАП последней генерирует в аналоговой абонентской линии напряжение, которое изменяется в соответствии с уровнем квантования. Главная сложность для клиентского модема заключается в необходимости восстановить из принятого сигнала ИКМ-коды, соответствующие каждому уровню напряжения, с частотой 8000 Гц.

Серверный модем подключается по цифровому каналу к цифровой КТСОП. Для кодирования сигналов серверного модема используются только те 256 кодов ИКМ, которые имеют место в цифровой части телефонной сети. Другими словами, отсутствует шум квантования, связанный с аналого-цифровым преобразованием. Эти ИКМ-коды преобразуются на АТС клиента в соответствующие аналоговые напряжения и отсылаются на клиентский модем по аналоговым линиям без потерь информации. Клиент принимает сигнал и восстанавливает исходные ИКМ-коды из аналогового сигнала.

K56-технология образует асимметричное соединение. Клиентский модем может принимать данные с большей скоростью, чем передавать их, так как при преобразованиях цифра-аналог информация не теряется. При посылке данных клиентским модемом (исходящий трафик), сигнал претерпевает преобразование аналог-цифра.

Теоретически технология ИКМ-модемов способна обеспечить скорость 64 кбит/с (передачу за каждую секунду 8000 символов по 8 бит в каждом), однако на практике такого быстродействия добиться невозможно по двум основным причинам:

1) первоначально в системах уплотнения использовались 8-битные АЦП/ЦАП, что при передаче сигнала, кодированного 256 уровнями (8 бит), приводило к появлению шума в последнем бите. Впоследствии появилась аппаратура уплотнения, в которой применяли АЦП/ЦАП большей разрядности, но использовали только 8 старших бит. Это позволило избежать появления шума в млад-

шем бите, но кое-где осталась аппаратура старого образца, поэтому надеяться на исчезновение шума не приходилось;

2) из-за того, что при разговоре по телефонной линии сигнал значительно меняет амплитуду (можно говорить шепотом, а можно кричать), для корректной передачи тихих звуков на входе аппаратуры уплотнения применяют нелинейное преобразование, а на выходе – обратное ему, что вызывает дополнительные шумы из-за неточности преобразований.

В результате этого было принято решение не использовать младший бит для передачи данных. Из-за этого теоретически возможная скорость соединения снизилась до 56 кбит/с. На самом деле и эта скорость практически не достижима, так как при работе на скорости 56 кбит/с пиковая мощность сигнала от модема провайдера превышает стандарты для телефонных линий. Модемы с таким превышением не допускают к использованию. Из-за снижения пиковой мощности сигнала до допустимых пределов максимальная скорость соединения снизилась до 53 кбит/с. Следовательно 56К-технологии не обеспечивают ожидаемое увеличение производительности по отношению к модемам 33,6 кбит/с.

В процессе установления связи серверный и клиентский модем «договариваются» между собой о количестве распознаваемых уровней напряжения при текущем состоянии абонентской линии. Скорость соединения по протоколу V.90 устанавливается от 32 до 56 кбит/с с шагом по 2 кбит/с.

Таким образом, К56-технология требует выполнения следующих условий:

- наличие цифрового канала в одном конце соединения, т. е. один конец соединения должен оканчиваться на магистральную цифровую линию (*trunk-side* T1, ISDN PRI или ISDN BRI). Локальные цифровые линии (line-side T1) не будут давать нужных результатов, так как на них будут иметь место дополнительные аналого-цифровые и цифро-аналоговые преобразования. В магистральных каналах сигнал преобразовывается лишь однажды, а после этого свободно достигает по цифровым каналам серверного модема;
- протокол V.90 должен поддерживаться модемами на обоих концах соединения: у клиента и на сервере провайдера;
- на пути от клиентского модема до серверного может быть только одно аналого-цифровое преобразование.

В табл. 2.5 показано, по какому протоколу взаимодействуют модемы различных технологий.

Таблица 2.5. **Совместимость модемов**

Клиент	Сервер			
	x2	K56flex	V.90	V.34
x2	56 K	V.34	56 K	V.34
K56flex	V.34	56 K	V.34	V.34
V.90	56 K	V.34	56 K	V.34
V.34	V.34	V.34	V.34	V.34

Технологии мобильной связи

Технология передачи данных по радиоканалу появилась довольно давно. По сравнению с технологией передачи данных по коммутируемым или выделенным каналам она имеет как преимущества, так и недостатки. Сети с использованием радиомодемов могут быть развернуты практически в любом географическом регионе. Радиус их действия составляет от десятков до сотен километров – в зависимости от мощности используемого приемопередатчика. Радиомодемы во многом похожи на обычные модемы для телефонных каналов. Основное отличие состоит в том, что они работают в канале множественного доступа (в данном случае единый радиоканал со многими пользователями), тогда как обычные проводные модемы – в канале типа «точка-точка». Алгоритмы работы сетей, использующих пакетные радиомодемы, описаны Рекомендацией AX.25.

Стандарт AX.25 устанавливает единый и обязательный для всех пользователей протокол обмена данными в пакетной радиосети. Он представляет собой специально переработанную для пакетных радиосетей версию стандарта X.25. В пакетных радиосетях используется канал множественного доступа. Протокол обмена AX.25 обеспечивает множественный доступ в канал связи с контролем занятости. Все пользователи сети считаются равноправными. Прежде чем начать передачу, радиомодем проверяет, свободен ли канал. В противном случае передача данных откладывается до освобождения канала. Одновременно с передачей данных радиомодем вырабатывает специальный сигнал, оповещающий остальные радиомодемы сети, что линия занята. При пакетной связи информация передается в виде отдельных блоков данных – кадров. Формат кадров стандарта AX.25, в основном, соответствует формату кадров протокола HDLC (High-level Data Link Control). Кадры бывают двух видов – служебные и информационные. Начало и конец кадра отмечаются флагами. Обычно поле флагов имеет вид – 01111110. Следующее поле, ADDRESS, содержит адреса отправителя, получателя, а также адреса станций-ретрансляторов, если таковые имеются. Как правило, станции-ретрансляторы используют при передаче на очень большие расстояния. Размер ADDRESS не должен превышать 70 байт. Поле CONT определяет вид кадра – служебный или информационный. Служебные кадры делятся на супервизорные и ненумерованные. Супервизорные кадры служат для подтверждения приема неискаженных кадров и запроса повторной передачи искаженных, а ненумерованные – для установления логического соединения при управлении обменом в сети. Информационные кадры содержат в себе передаваемую информацию, находящуюся в поле INFORM. Размер его, как правило, не больше 256 байт (с увеличением размера сильно повышается вероятность ошибок при передаче). Поле CRC служит для обнаружения ошибок. Его размер зависит от разрядности CRC, например, для CRC-16 размер поля составляет 2 байт.

Как правило, радиомодем представляет собой прямоугольный ящик, внешне очень напоминающий видеомагнитофон. На передней панели находится дис-

плей и кнопки управления, благодаря чему управлять работой радиомодема можно «вручную». Его вес варьируется от 1 до 5 кг. Пакетный радиомодем состоит из модема и пакетного контроллера TNC (Terminal Node Controller). Именно TNC выполняет основные функции, такие, как форматирование кадров и доступ к радиоканалу множественного доступа, кодирование и т.д. В общем случае станция в пакетной радиосети включает компьютер, радиомодем и радиостанцию КВ- или УКВ-диапазона. Однако на практике возможны вариации. Например, учитывая, что TNC – высокоинтеллектуальное устройство, вместо компьютера можно использовать простой терминал. Кроме того, радиомодемы могут соединять сегменты сети, построенные на основе кабеля. При работе в диапазоне коротких волн используется частотная модуляция в полосе частот телефонных каналов КТСОП. При этом перепад частот всегда равен 200 Гц. В Европе обычно используется частота 1850 Гц для передачи «0» и 1650 Гц для передачи «1». Скорость передачи при использовании КВ невелика и, как правило, не превышает 300 бит/с. В УКВ-диапазоне можно достичь намного больших скоростей. Перепад поднесущих частот здесь равен 1000 Гц, «0» соответствует частота 1200 Гц, а «1» – 2300 Гц. При использовании относительной фазовой модуляции скорость передачи может достигать 19200 бит/с.

Портативный ПК пользователя оснащают пакетным радиомодемом, который разбивает поток данных на небольшие цифровые пакеты. Радиомодем связывается с радиосетью, а поставщик услуг обеспечивает передачу этих пакетов по назначению. На противоположном конце канала другой пакетный радиомодем принимает пакеты и передает их программному обеспечению ПК. Доступная цена средств пакетной радиосвязи – это одно из наиболее привлекательных достоинств данной технологии.

Пакетная радиосвязь обладает рядом преимуществ. Во-первых, нет необходимости в каналах связи, создание радиосети актуально при отсутствии развитой инфраструктуры связи; во-вторых, пакетные радиосети имеют неплохую масштабируемость и гибкость (расположение станций может постоянно меняться), и в-третьих, исключается возможность любых «обрывов на линии». Пакетные радиосистемы оптимальны для небольших объемов передаваемой информации (передача документов, справок, выписок). Денежные средства и время, необходимые для создания пакетных радиосетей, как правило, намного меньше, чем для обычных кабельных сетей. Рассматриваемая технология практически избавляет пользователя от забот по обеспечению информационной безопасности, поскольку поставщик услуг может обеспечить шифрование данных, используя алгоритмы различных типов. Однако для достижения наибольшей безопасности необходимо, чтобы приложение пользователя само шифровало данные, например, по технологии с открытым ключом.

Пакетная радиосвязь имеет и недостатки. Наиболее значимые – малая скорость передачи данных, отсутствие общепринятых стандартов и сложность эксплуатации (подключение TNC к компьютеру и к радиостанции с последующим конфигурированием). Данные передаются на скорости от 4,8 до 19,2 кбит/с,

но верхнего ее предела можно достичь не во всех регионах, в которых действуют системы, реализующие эту технологию. Отсутствие же общепринятых стандартов привело к тому, что различные поставщики услуг применяют разные протоколы передачи данных, а в результате их коммуникационные инфраструктуры становятся частично или полностью несовместимыми. В большинстве случаев, чтобы использовать службы другого поставщика, пакетный радиомодем надо заменять. Технология пакетной радиосвязи наилучшим образом подходит для приложений электронной почты или быстрых операций, например таких, как транзакции с кредитными карточками. Поскольку пакетная радиосвязь не обеспечивает прямого канала по схеме «точка-точка», то для передачи файлов она неэффективна. Еще одним ее недостатком является поддержка весьма ограниченного числа коммуникационных средств прикладного уровня.

Сотовая связь с коммутацией каналов использует существующие аналоговые сотовые сети, только в отличие от пакетной коммутации в ней вместо коммутации пакетов данных используется обычная коммутация каналов сотовой сети. Для передачи данных пользователь подключает *сотовый модем* к своему ПК и сотовому телефону, поддерживающему передачу данных, и устанавливает коммутируемое соединение точно так же, как при работе с аналоговым модемом.

У сотовых модемов много общего со стандартными портативными модемами: они выпускаются как в стандарте на средства расширения портативных ПК (*PCMCIA*), так и для внешнего подключения и могут устанавливать соединения при помощи обычных аналоговых телефонных линий. Существенным свойством сотового модема является его способность выйти за рамки передачи по кабельным линиям, формируя и поддерживая качественные соединения в сотовой сети. Это осуществляется за счет поддержки протокола исправления ошибок сотовой связи, например ETC (Enhanced Throughput Cellular) компании AT&T Paradyne или MNP (Microcom Network Protocol) – 10 фирмы Microcom, управляющего передачей сигналов в сотовой среде. Сотовая связь с коммутацией каналов – довольно медленный вид связи: данные передаются на скоростях до 14,4 кбит/с и лишь в отдельных зонах обслуживания скорость увеличивается до 20 кбит/с. В крупных городах и при удалении от базовой станции скорость передачи может снижаться.

2.4. Технические средства телекоммуникаций

Связные интерфейсы

Связные интерфейсы (СИ) используют для подключения ЭВМ или терминала с сетью передачи данных (СПД). Для описания СИ с сетевой стороны используют аббревиатуру АПД (аппаратура передачи данных) или DCE (Data Circuit terminating Equipment). Для описания связного интерфейса со стороны ЭВМ или терминала ООД (оконечное оборудование данных) или DTE (Data Terminal Equipment). ООД работает под управлением программно-аппаратных средств и входит в состав ЭВМ или терминала. Аппаратные средства,

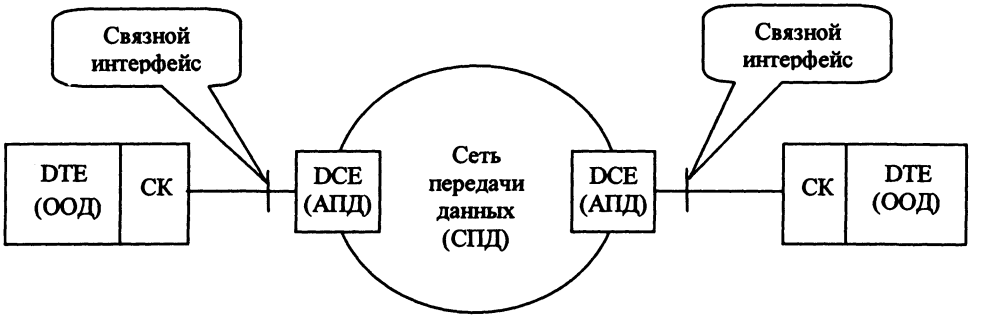


Рис. 2.27. Подключение ООД к сети

называемые «связной контроллер» (СК), конструктивно могут принимать различные формы. СПД обычно работают в последовательном режиме, и поэтому СК должен обеспечить среди прочих такие функции, как преобразование данных, передаваемых в сеть из параллельной формы представления в последовательную и обратно (рис. 2.27).

Связной контроллер, являющийся частью ЭВМ, строится по модульному принципу: «главный» аппаратный модуль обеспечивает обмен данными между ЭВМ и «вторичными» аппаратными модулями; «вторичные» аппаратные модули, подключенные к «главному» модулю, обеспечивают обмен данными между связным интерфейсом и «главным» модулем. В мини-ЭВМ «главный» модуль используют не всегда. В ПК он отсутствует. В этом случае «вторичные» аппаратные модули подключают непосредственно к ЭВМ.

«Вторичные» модули бывают двух типов:

- работающие с асинхронными СИ;
- работающие с синхронными СИ.

Обычно каждый «вторичный» аппаратный модуль называемый Аппаратный связной интерфейс (АСИ) выполняют в виде печатной платы, обеспечивающей подключение от 1 до 16 СИ.

Асинхронный АСИ передает и принимает последовательные данные в асинхронном, стартстопном формате. Чаще используют линейный код МТК-5 (IA5 International Alphabet №5) эквивалентный ASCII (7 бит + 1 бит проверки на четность).

Бит проверки на четность может принимать значения: *N* – нет контроля (*NONE*), *O* – сумма нечетная (*ODD*), *E* – сумма четная (*EVEN*), *M* – всегда единица (*MARK*), *S* – всегда ноль (*SPACE*).

Типичные скорости асинхронной передачи приведены в табл. 2.6.

Синхронный АСИ посылает и принимает последовательные данные через сеть в виде блоков (кадров), при этом каждый блок (кадр) представляет собой непрерывный, последовательный поток данных. Как уже отмечалось, при синхронной передаче каждый передаваемый знак состоит из 8 бит (стоповые биты отсутствуют). Временные промежутки между знаками в блоке не допускаются.

Таблица 2.6. **Скорости асинхронной передачи**

Скорость передачи, бит/с	Знаковая скорость, зн/с	Число бит в знаке	Скорость передачи, бит/с	Знаковая скорость, зн/с	Число бит в знаке
50	6,6	7,5	600	60	10
75	10	7,5	1200	120	10
100	10	10	1800	180	10
110	10	11	2400	240	10
150	15	10	4800	480	10
200	20	10	9600	960	10
300	30	10	19200	1920	10

Данные, посылаемые АСИ, обычно синхронизируются в соответствии с сигналом синхронизации, поступающим из сети через интерфейсный кабель. Это имеет свои преимущества. Не меняя настройки АСИ, можно повысить скорость передачи (бит/с) по сетевому каналу, заменив низкоскоростной модем высокоскоростным. Типичные скорости синхронной передачи приведены в табл. 2.7.

Таблица 2.7. **Скорости синхронной передачи**

Скорость передачи, бит/с	Знаковая скорость, зн/с	Скорость передачи, бит/с	Знаковая скорость, зн/с
1200	150	16000	2000
2400	300	16800	2100
4800	600	19200	2400
7200	900	48000	6000
9600	1200	56000	7000
12000	1500	64000	8000
14400	1800	72000	9000

Характеристики аппаратного связного интерфейса. АСИ (ООД), подключенный к сети (АПД), должен быть совместим по своим характеристикам с СИ и сетью (АПД). Каждый провод в многопроводном цифровом интерфейсе называется «цепью обмена». Цепи обмена используют для передачи данных, управления и синхронизации.

Основными характеристиками АСИ являются: скорость передачи, электрические характеристики цепей обмена, функциональное назначение цепей обмена, процедурные и механические характеристики интерфейса. Его скорость передачи определяет в конечном счете возможности ООД этой сети.

Электрические характеристики цепей обмена определяют уровни напряжений, используемых для представления данных и управляющих сигналов, передаваемых через интерфейс; допустимые значения фронтов сигналов, затухания сигналов; допустимую нагрузку на каждую цепь и другие электрические параметры соединения.

Каждая цепь обмена интерфейса имеет свое название и функциональное назначение, определяемое направлением передачи данных или управляющего сигнала и выполняемые действия. Процедурные характеристики интерфейса определяют последовательность обмена управляющими сигналами и данными, передаваемыми через интерфейс. Механические характеристики определяют геометрические размеры разъемов интерфейса, их контактов, вид механического соединения, обеспечивающего передачу сигналов через разъемное соединение.

Стандарты аппаратных связных интерфейсов. CCITT (Consultative Committee on International Telephony and Telegraphy, Международный телеграфный и телефонный консультативный комитет – МККТТ), в английской нотации называемый ITU-T (International Telecommunications Union – Technical Standards Sector, Международный телекоммуникационный союз – Сектор технических стандартов) – это международная организация, создающая стандарты для телекоммуникаций. Их проекты являются основными рекомендациями для всех остальных в этой области. Те рекомендации, которые относятся к применениям модемов, имеют префикс «V» и называются рекомендациями серии V.

RS-232C (V.24/V.28). Это рекомендованный стандарт (RS –Recommended Standard) EIA (Electrical Industry Association – Ассоциация электрической промышленности), определяющий последовательный коммуникационный интерфейс (т. е. способ взаимодействия) между DTE и DCE. Число 232 – исходный серийный номер данного стандарта. Наиболее часто используют вариант «*C*» этого стандарта, т. е. RS-232C. В случае, когда используют вариант «*D*», префикс RS меняется на EIA. Кроме нескольких дополнительных (но редко используемых) сигналов, между вариантами «C» и «D» практически нет никакой разницы.

Стандарт RS-232C описывает несимметричный интерфейс между ООД и АПД, работающий в режиме последовательного обмена двоичными данными со скоростями передачи до 20000 бит/с.

Интерфейс называется *несимметричным*, если для всех цепей обмена этого интерфейса используется один общий возвратный провод – сигнальная «земля» (рис. 2.28). В *симметричных* интерфейсах собственную сигнальную «землю» имеет каждая цепь обмена. Как видно из рис. 2.28, напряжения сигналов в цепях обмена симметричны по отношению к уровню сигнальной «земли» и составляют не менее +3В для двоичного нуля и не более –3В для двоичной единицы. Напряжения вне этих диапазонов порождают неопределенное состояние. На практике фактически используемые уровни зависят от источников напряжений, подаваемых на схемы интерфейса, и могут достигать ±12В.

Несимметричный интерфейс работает в асинхронном и синхронном режимах передачи данных. Длина кабеля ограничена 15 м, скорость передачи не более 19200 бит/с, хотя при присоединении периферийных устройств к компьютерам эти значения могут быть превышены. Стандарт EIA RS-232C эквивалентен:

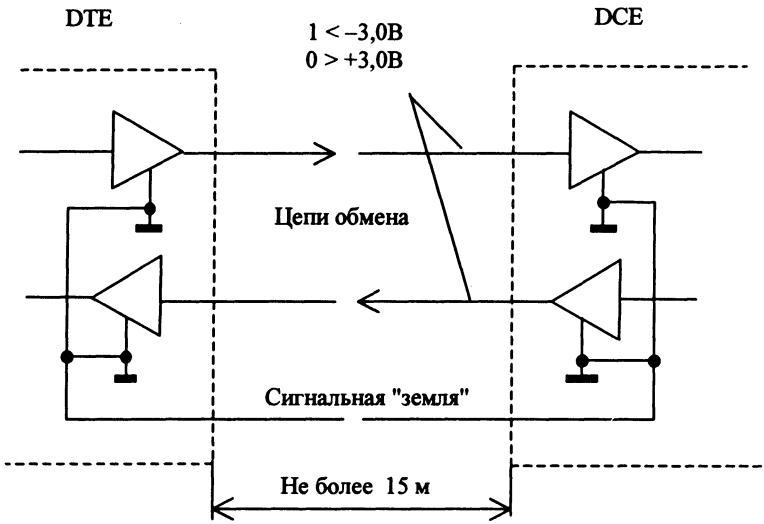


Рис. 2.28. Интерфейс RS-232C

• по функциональному и процедурному описанию цепей обмена рекомендации V.24 ITU-T;

• по характеристикам электрического сигнала рекомендации V.28 ITU-T;

• по механическим характеристикам рекомендации ISO 2110 (25 контактный разъем – DB25). На аппаратуре DTE (в том числе, и на COM-портах ПК) принято устанавливать *вилки* DB25-P или более компактный вариант – DB9-P. Девятиштырьковые разъемы не имеют контактов для дополнительных сигналов, необходимых для синхронного режима (в большинстве 25-штырьковых разъемов эти контакты не используются). На аппаратуре DCE (модемах) устанавливают *розетки* DB25-S или DB-9S. Это правило предполагает, что разъемы DCE могут подключаться к разъемам DTE непосредственно или через переходные «прямые» кабели с розеткой и вилкой, у которых контакты соединены «один в один». Переходные кабели могут являться и переходниками с 9- на 25-штырьковые разъемы.

В качестве контроллера АСИ в ПК используют контроллер коммуникационного порта UART (Universal Asynchronous Receiver Transmitter – Универсальный асинхронный приемо-передатчик). Это устройство применяют в DTE или DCE для получения и передачи асинхронных данных. Устройство UART, используемое в современных ПК (типа NS16550A с внутренним буфером данных) обеспечивает скорость до 115200 бит/с.

Основные сигнальные цепи для асинхронной связи с модемом приведены в табл. 2.8.

Таблица 2.8. **Сигнальные цепи для асинхронной связи**

Обозначение цепи		Контакт разъема		Направление	Название цепи
RS232	V.24	DB25S	DB9S	I/O	
PG	101	1	–	–	Protect Ground – Защитная «земля»
TD	103	2	3	O	Transmit Data – Передаваемые данные
RD	104	3	2	I	Receive Data – Принимаемые данные
RTS	105	4	7	O	Request To Send – Запрос на передачу
CTS	106	5	8	I	Clear To Send – Готовность модема к приему данных для передачи
DSR	107	6	6	I	Data Set Ready – Готовность модема к работе
SG	102	7	5	–	Signal Ground – Схемная земля
DCD	109	8	1	I	Data Carrier Detect – Несущая обнаружена
DTR	108/2	20	4	O	Data Terminal Ready – Готовность терминала (DTE) к работе
RI	125	22	9	I	Ring Indicator – Индикатор вызова

Если аппаратура DTE соединяется без модемов, то разъемы устройств (вилки) соединены между собой *нуль-модемным кабелем*, имитирующим сетевое соединение и имеющим на обоих концах розетки, контакты которых соединены по одной из схем, приведенных на рис. 2.29.

Назначение регистров контроллера коммуникационного порта RS-232C (V.24/V.28). Управление переносом информации по интерфейсу RS-232 выполняет специальная электронная схема (обычно UART – NS16550A), объединяющая до четырех контроллеров, каждый из которых обслуживает одну линию связи. Физически компьютер может иметь и меньшее количество контроллеров

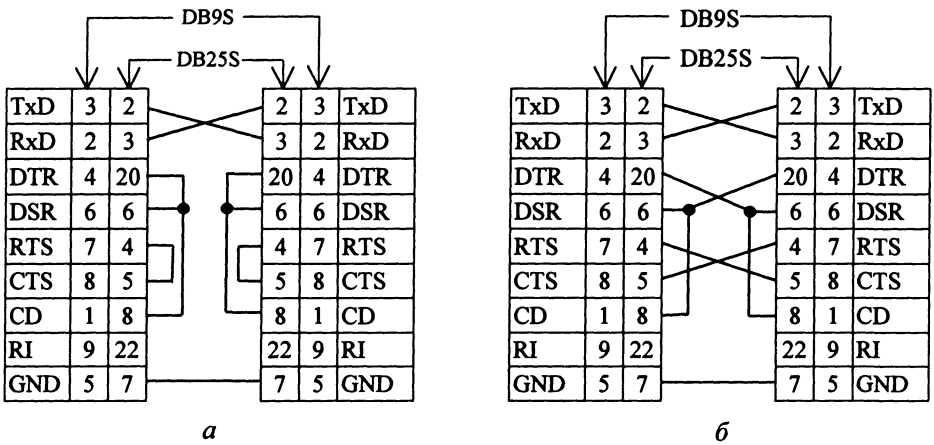


Рис. 2.29. Нуль-модемный кабель:
а – минимальный, *б* – полный кабель

последовательной связи. Контроллер способен как передавать, так и принимать информацию из интерфейса. Его особенностью является способность генерировать аппаратные прерывания по программируемой маске условий. IBM PC поддерживает аппаратно генерацию прерываний для последовательных контроллеров COM1 и COM2. COM1 генерирует прерывание *0Ch*, контроллер COM2 – прерывание *0Bh*. Каждый контроллер управляет десятью внутренними регистрами, доступ к которым осуществляется по номерам портов. Они отсчитываются от базового адреса (их можно получить из области данных BIOS). Одни и те же адреса портов могут соответствовать различным внутренним регистрам в зависимости от направления передачи информации и значения бита 7-го регистра управления линией. Рассмотрим регистры подробней.

Регистр данных (DR, базовый адрес+0). Использование регистра данных зависит от состояния регистра управления линией и выполняемой операции доступа к порту (чтение или запись). Если контроллер передает или принимает информацию, в бите 7-го регистра LCR должен быть записан 0. В это случае запись байта в порт DR помещает данные в регистр данных передатчика. Отсюда они без каких-либо дополнительных управляющих воздействий передаются в сдвиговый регистр, который формирует слово информации и бит за битом передает его в интерфейс. Чтение байта из внутреннего регистра DR возвращает последний принятый контроллером байт данных.

Регистры делителя частоты (DLLBR и DLHBR, базовый адрес+0 и базовый адрес+1). Если бит 7 LCR установлен в 1, содержимое этого регистра определяет формат данных интерфейса. В этом случае порты регистров DLLBR и DLHBR используются как защелки для так называемого делителя частоты приема/передачи. Значение регистра – это число, полученное делением числа 1843200 на скорость передачи в бодах, умноженную на 16. Делитель частоты равный 1, соответствует максимально возможной скорости передачи в 115200 бод.

Регистр разрешения прерываний (IER, базовый адрес+1). Аппаратура контроллера последовательной связи может генерировать запросы аппаратных прерываний при возникновении в контроллере одной или нескольких ситуаций четырех возможных типов:

Бит	Назначение
7–4	Не используются
3	Разрешена генерация прерываний при изменении значения регистра состояния модема (MSR)
2	Разрешена генерация запроса прерывания при изменении значения регистра состояния линии (LSR)
1	Разрешена генерация запроса прерывания в случае, когда пуст регистр данных передатчика
0	Разрешена генерация запроса прерывания в случае готовности принятых данных

Запись в регистр IER байта 0 запрещает генерацию запроса прерывания для всех возможных ситуаций, а запись *0Fh* разрешает генерацию запросов прерываний всех возможных типов ситуаций.

Регистр идентификации прерывания (IIR, базовый адрес+2). Это только читаемый внутренний регистр, позволяющий программе-обработчику прерывания определить причину, по которой сгенерировано прерывание или, другими словами, причину, по которой программа получила управление.

Бит	Назначение
7–3	Не используются
2–1	Идентификатор причины прерывания: 00 – прерывания по причине изменения регистра состояния модема 01 – пуст регистр данных передатчика 10 – готовность полученных данных 11 – прерывание по причине изменения регистра состояния линии: либо установка бита ошибки, либо выделено условие BREAK
0	1 – нет активных прерываний 0 – есть ожидающее обработки прерывание

Устранение причины возникновения прерывания осуществляется чтением или записью соответствующего случаю содержимого регистра. Возможно одновременное появление причин для разных типов прерываний, и в этом случае аппаратура контроллера, выполняющая генерацию прерываний, упорядочивает их по приоритету. Прерывание, связанное с изменением регистра состояния модема всегда имеет наивысший приоритет. Следующим по приоритету является прерывание по готовности принятых данных. После него генерируется прерывание из-за опустевшего регистра данных передатчика. Наименьший приоритет имеет прерывание по изменению состояния линии.

Одному прерыванию может соответствовать несколько причин одновременно, и устранение какой-либо одной из них, тем не менее, не вызовет установки в «1» бита 0 регистра идентификации прерывания. Поэтому обработчик не должен завершать свое выполнение до устранения всех причин возникновения прерывания.

Регистр управления линией (LCR, базовый адрес+3). Этот регистр используется для задания формата слова, передаваемого по интерфейсу, и некоторых дополнительных особенностей передачи или приема.

Бит	Назначение
7	Выбор содержимого в портах базовый адрес+0 и базовый адрес+1
6	0 – обычное функционирование контроллера; 1 – посылка сигнала *BREAK*;
5–3	Выбор способа контроля по четности : xx0 – отсутствие бита контроля по четности (*N*); 001 – бит контроля формируется по четному паритету (*E*);

	011 – бит контроля формируется по нечетному паритету (*O*);
	101 – бит контроля равен 1 (*M*);
	111 – бит контроля равен 0 (*S*)
2	Число стоп-битов
	0 – 1 стоп-бит;
	1 – 2 стоп-бита
1–0	Длина слова
	00 – 5 бит;
	01 – 6 бит;
	10 – 7 бит;
	11 – 8 бит

Регистр управления модемом (MCR, базовый адрес+4). Данный регистр управляет работой контроллера последовательной связи.

Бит	Назначение
7–5	Не используются
4	Запуск автономного теста контроллера последовательной связи
3	Разрешает прерывания от контроллера
2	Определяемый пользователем запрос на прерывание
1	Устанавливает сигнал на линии интерфейса *RTS*
0	Устанавливает сигнал на линии интерфейса *DTR*

Регистр состояния линии (LSR, базовый адрес+5). Этот регистр сообщает компьютеру информацию о состоянии тракта приема-передачи контроллера.

Бит	Назначение
7	Всегда 0
6	Сдвиговый регистр передатчика пуст
5	Буферный регистр передатчика пуст
4	Принят *BREAK*
3	Ошибка кадрирования (нарушение синхронизации приемника и передатчика)
2	Ошибка четности
1	Ошибка переопределения данных
0	Готовность принятых данных

Регистр состояния модема (MSR, базовый адрес+6). При использовании модема (или нуль-модемного соединения), компьютер, опрашивая этот регистр, отслеживает все изменения состояния модема, что позволяет определить возможность передачи байта (наличие сигнала CTS), необходимость приема информации (наличие сигнал DCD) и т. п. Четыре старших бита регистра показывают уровень напряжения на соответствующей линии интерфейса RS-232, а четыре оставшихся – наличие изменения уровня сигнала с момента последнего чтения содержимого, так как любое чтение обнуляет эти биты.

Бит	Назначение
7	Сигнал на линии *DCD*
6	Сигнал на линии *RI*
5	Сигнал на линии *DSR*
4	Сигнал на линии *CTS*
3	Изменение на линии *DCD*
2	Изменение на линии *RI*
1	Изменение на линии *DSR*
0	Изменение на линии *CTS*

Существуют две стратегии организации обмена по интерфейсу RS-232:

- управляемый прерываниями обмен данных;
- последовательный опрос.

В первом случае контроллер инициализируется так, что те или иные события в контроллере или линиях интерфейса генерируют аппаратные прерывания. Эти прерывания обслуживает программа-обработчик, которая принимает очередной символ, помещает его в буфер или передает очередной байт в интерфейс. Основная программа читает байты уже из памяти, а при необходимости передачи байта записывают его в передающий буфер.

Во втором случае компьютер выполняет бесконечный цикл опроса регистров контроллера, ожидая наступления некоторых событий.

Интерфейс RS-449A. В отличие от RS-232C интерфейс RS-449A имеет две отдельные электрические спецификации:

несимметричный электрический интерфейс RS-423A;

симметричный электрический интерфейс RS-422A.

Если физическую удаленность и скорость передачи, определенные стандартом RS-232C, необходимо увеличить, то следует использовать альтернативные типы сигналов, определенные в стандарте RS-422A (V.11). В нем применяют сочетание кабеля из витой пары и дифференциальных передающей и принимающей схем (рис. 2.30). Дифференциальная передающая схема вырабатывает два равных, но противоположной полярности сигнала для каждого подлежащего передаче бита. Так как дифференциальная принимающая схема чувствительна только к разности между двумя сигналами на двух своих входах, то любой шум, воздействующий на оба провода, не окажет влияния на работу приемника. Поэтому говорят, что дифференциальные приемные устройства обладают высокой помехоустойчивостью.

Стандарт RS-423A является производным от RS-422A и может быть использован для приема выходного напряжения с помощью дифференциального принимающего устройства согласно интерфейсу RS-232C. Стандарт RS-422A в сочетании с кабелем из витой пары предпочтителен для расстояний порядка 100 м при скорости в 1 Мбит/с или для больших расстояний, но при меньших скоростях (табл. 2.9). Важным показателем любой линии передачи является ее характеристическое сопротивление (*импеданс*) – Z_0. Приемное устройство

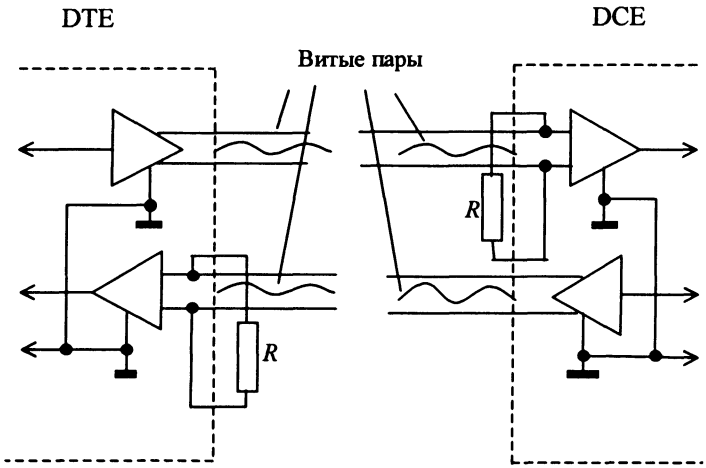


Рис. 2.30 Интерфейс RS-422A

поглощает (адсорбирует) все поступившие сигналы, если линия заканчивается согласующим сопротивлением, равным Z_0. В противном случае возникает *отражение сигнала*, которое, в свою очередь, вызывает дальнейшее искажение поступивших сигналов. Поэтому линии передачи, как правило, заканчиваются согласующим сопротивлением (R = 50...200 Ом).

Механические характеристики, определяемые стандартом ISO 4902 (37-контактный разъем и 9-контактный разъем для вспомогательного канала связи). Зависимость максимальной скорости передачи от длины кабеля для RS-449A приведена в табл. 2.9.

Таблица 2.9. **Зависимость скорости передачи от расстояния для интерфейса RS-449A**

Интерфейс	Расстояние, м		
	10	100	1000
RS -423A	100	10	1
RS-422A (несогласованный)	1 000	100	10
RS-422A (согласованный)	10 000	1000	100

Интерфейс V.35. Этот стандарт определяет синхронный интерфейс для работы с аналоговым широкополосным модемом со скоростью 48 кбит/с. Каждый широкополосный модем устанавливает связь через широкополосный аналоговый канал, используя полосу частот от 60 до 108 кГц, что эквивалентно полосе частот, занимаемой 12 телефонными каналами. Интерфейс V.35 использует комбинацию несимметричных (V.24/V.28) и симметричных (V.35) сигна-

лов, и поэтому максимальная длина интерфейсного кабеля та же, что и для интерфейса RS-232C – 15 м. Интерфейсный разъем V.35 имеет 34 контакта по стандарту ISO 2593.

Модемы

Модем (МОдулятор-ДЕМодулятор) – устройство преобразования последовательных цифровых сигналов в аналоговые и наоборот. Организации по стандартизации используют общепринятые аббревиатуры АПД (DCE) для обозначения модема и ООД (DTE) для обозначения ЭВМ, терминала или любого другого устройства, подключенного к модему.

Модем имеет два интерфейса (рис. 2.31): интерфейс между DCE и аналоговой линией; многопроводный цифровой интерфейс между DCE и DTE.

Двухточечный канал. Простейшей сетью с использованием модемов, является двухточечный канал, в котором два модема соединены («точка-точка») одной линией связи (рис. 2.32). Дискретный канал соединяет DTE с DTE. Линия соединяет DCE с DCE. Дискретный канал состоит из линии и двух модемов (DCE). При скорости передачи до 20 кбит/с используют интерфейс V.24/V.28 (RS-232C), осуществляемый с помощью 25- или 9-контактного гнездового разъема. При скоростях передачи от 48 до 168 кбит/с необходимы широкополосные модемы, работающие с интерфейсом V.35. При скоростях до 20 кбит/с может быть использована любая из следующих аналоговых телефонных линий связи:

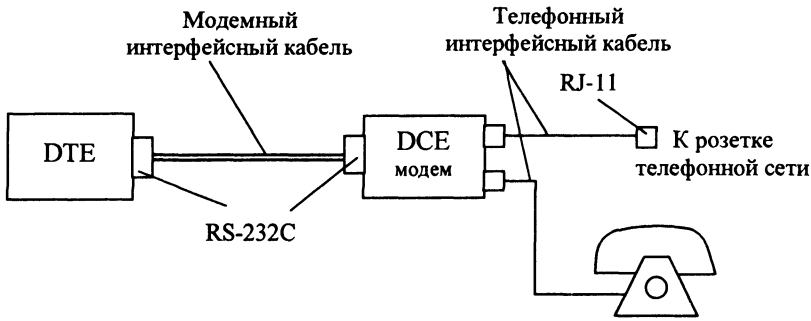


Рис. 2.31. Интерфейсы модема

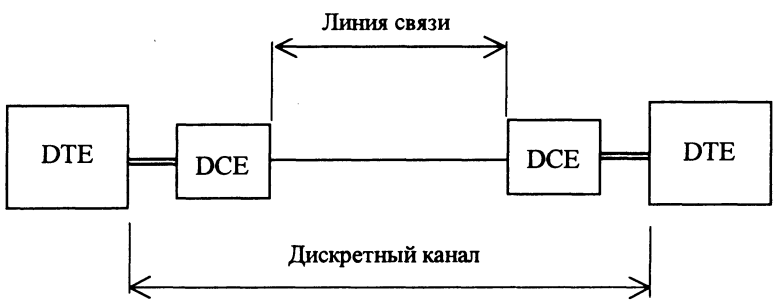


Рис. 2.32. Двухточечный канал

- 4-проводная 2-точечная выделенная линия;
- 4-проводная многоточечная выделенная линия;
- 2-проводная 2-точечная выделенная линия;
- 2-проводная 2-точечная коммутируемая линия (связь путем набора номера через КТСОП);
- 4-проводная 2-точечная коммутируемая линия, организуемая путем коммутации двух отдельных двухпроводных соединений через КТСОП.

Стандарты телефонных каналов как производные от стандартного канала КТСОП тональной частоты (ТЧ) представлены в табл. 2.10.

Таблица 2.10. **Стандарты телефонных каналов**

Канал	Количество каналов ТЧ	Полоса частот, кГц
Стандартный ТЛФ канал	1	0.3...3.4
Первичный широкополосный	12	60...108
Вторичный широкополосный	60	312...552
Третичный широкополосный	300	812...2044
Четверичный широкополосный	900	8516...12338

Режимы работы модемов. *Асинхронный.* Данный режим реализуется асинхронными модемами, такие модемы являются низкоскоростными и работают в режиме асинхронной стартстопной позначной передачи. Асинхронные модемы не генерируют сигналов синхронизации и могут работать с любой скоростью передачи в пределах установленного для них диапазона скоростей.

Синхронный. В этом режиме данные передаются блоками, а модем генерирует сигналы синхронизации. Модемы, реализующие только синхронный режим, называются синхронными модемами.

Асинхронно-синхронный. Такой режим реализуется асинхронно-синхронными модемами, которые могут осуществлять как синхронную, так и асинхронную передачу. Модем удаляет стартстопные биты перед передачей и восстанавливает их после приема. Модемы этого типа генерируют сигналы синхронизации и имеют встроенный асинхронно-синхронный преобразователь. Асинхронно-синхронные и синхронные модемы работают только с фиксированными скоростями передачи.

При выборе модема важное значение имеет тип связи, обеспечиваемый комбинацией модема с линией. Любой модем, работающий с 4-проводной 2-точечной линией, использует одну пару для передачи, а вторую для приема и, следовательно, может работать в дуплексном режиме. Модемы, работающие с 4-проводной многоточечной линией работают только в полудуплексном режиме. Модемы, имеющие только синхронный режим, работают на 4-проводной 2-точечной некоммутируемой линии, либо через КТСОП, при этом одно коммутируемое соединение обеспечивает полудуплексный режим, а двойное коммутируемое соединение – дуплексный режим.

Асинхронно-синхронные модемы работают на 2-проводных линиях (либо выделенных, либо коммутируемых), и все они могут работать в дуплексном режиме.

Совместимость модемов. Передачу данных по телефонным сетям описывают рекомендации серии V Международного телекоммуникационного союза (Сектор технических стандартов) – ITU-T. Проверкой совместимости является проверка номера серии V, указанного фирмой-изготовителем в спецификациях модема. Классификация рекомендаций серии V приведена на рис. 2.33.

Модем может работать в двух режимах: командном и передачи данных. Командный режим модема, как правило, устанавливается:

при включении питания;

при первоначальной инициализации модема;

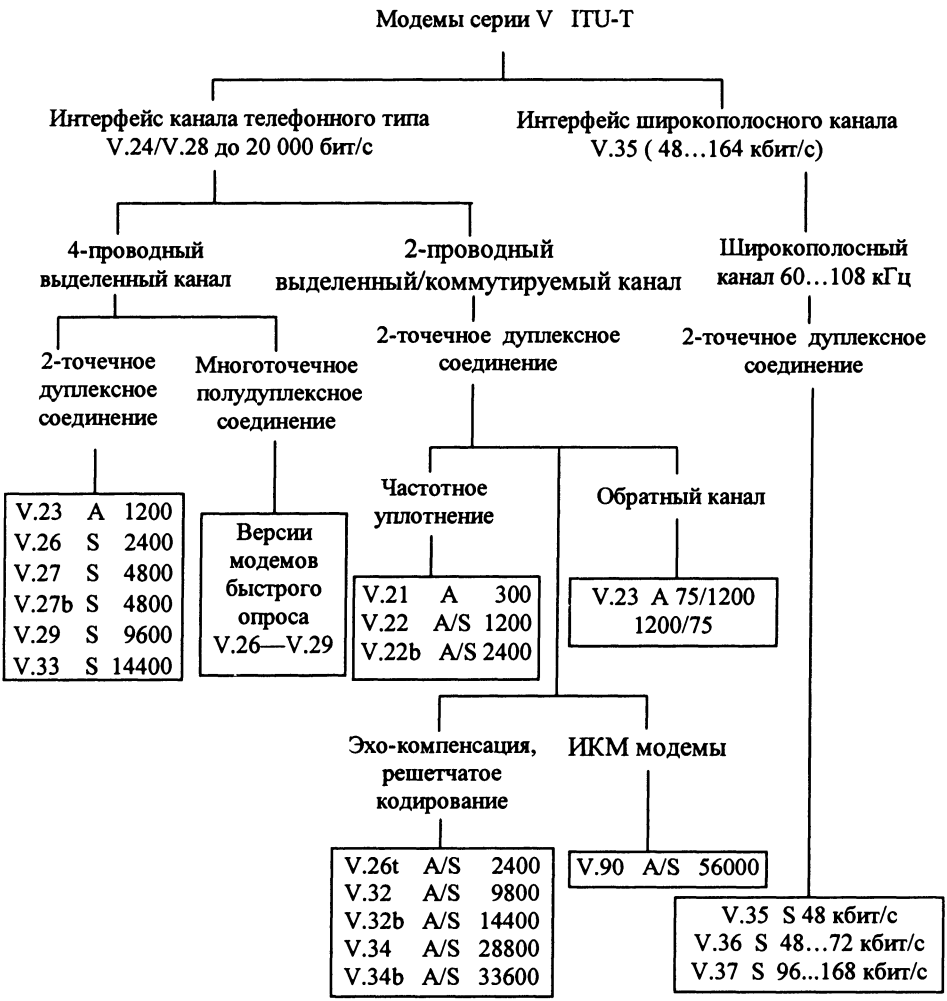


Рис. 2.33. Классификация рекомендаций серии V для модемов

после неудачной попытки соединения с удаленным модемом;

при прерывании с клавиатуры нажатием комбинации клавиш «положить трубку» (чаще всего <Alt><H>);

при выходе из режима передачи данных через ESCAPE-последовательность.

В командном режиме весь поток данных, поступающий в модем через интерфейс V.24/V.28, воспринимается им как команда.

Режим передачи данных (on-line) устанавливается после посылки модемом сообщения CONNECT в случаях:

при удавшейся попытке установления связи с удаленным модемом;

при выполнении модемом самотестирования.

В режиме передачи данных поток данных, поступающий в модем из DTE транслируется с преобразованием в линию, а поток данных из линии транслируется с обратным преобразованием в интерфейс с DTE.

Функциональные режимы модема. Модем всегда находится в одном из двух функциональных режимах (за исключением периодов, когда он переходит из одного режима в другой): командном (локальном) и в режиме асинхронного соединения (ON LINE). Схема переходов модема представлена на рис. 2.34. При включении питания модем инициализирует свои параметры в соответствии с конфигурацией, записанной в энергонезависимой памяти, и переходит в асинхронный командный режим. Только в этом режиме модем воспринимает АТ-команды. По Z-команде модем восстанавливает свою рабочую конфигурацию

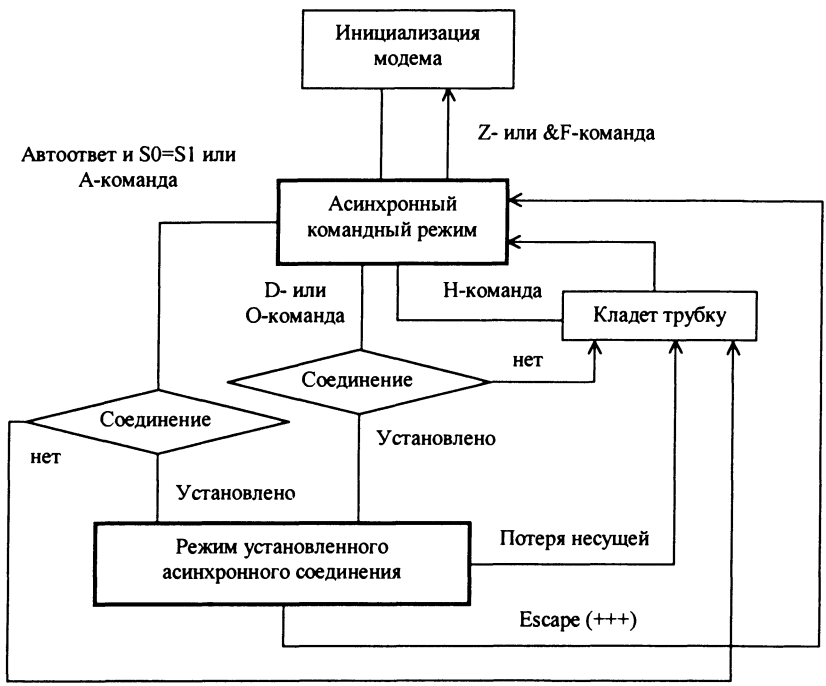


Рис. 2.34. Граф состояния модема

из энергонезависимой памяти и возвращается в командный режим. *&F*-команда восстанавливает конфигурацию по профайлу фирмы-изготовителя (установка по умолчанию) и возвращается в командный режим. Модем «поднимает трубку» в режиме автоответа:

а) при поступлении A-команды;

б) автоматически при S1 = S0, когда счетчик поступивших звонков (вызовов) становится равным числу, установленному для ответа;

в) при поступлении команды набора номера, когда строка вызова заканчивается R.

Функции цепей обмена 103, 104, 109 V.24. Рассмотрим функции цепей обмена, связанные с передачей и приемом данных:

103 (2) TxD (передаваемые данные) к DCE;

104 (3) RxD (принимаемые данные) к DTE;

109 (8) CD (детектор принимаемого линейного сигнала) к DTE.

Входной поток последовательных данных, поступающих в модем через цепь 103, преобразуется модулятором в модулированный аналоговый сигнал для вывода его в линию (рис. 2.35). На другом конце линии демодулятор удаленного модема принимает модулированный линейный сигнал и преобразует его в поток последовательных данных для вывода через цепь приема данных 104.

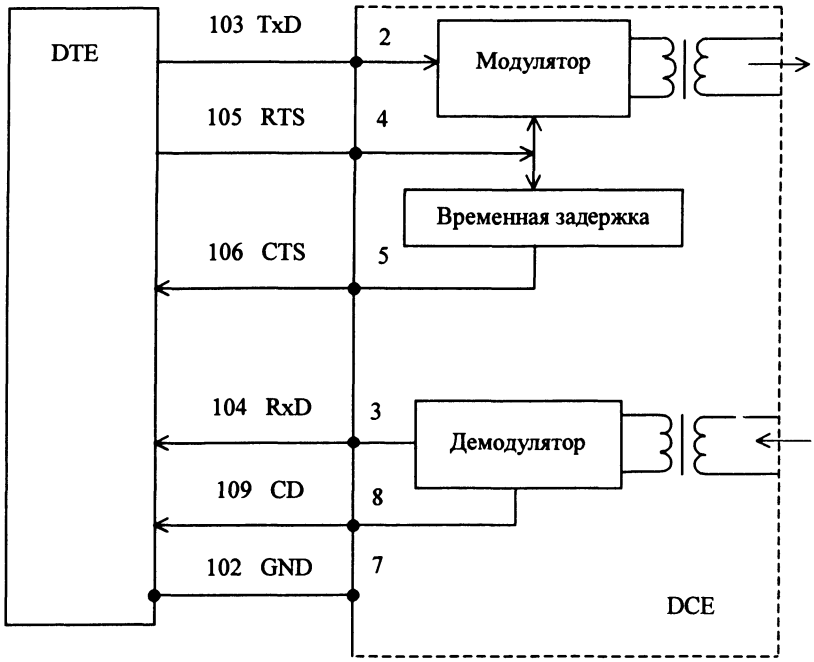


Рис. 2.35. Функции цепей обмена 103, 104, 109

При обнаружении модулированной несущей частоты демодулятором цепь 109 переходит из состояния ВЫКЛ в состояние ВКЛ. При этом между моментом обнаружения несущей и моментом изменения состояния цепи обмена 109 вносится задержка, известная как задержка «включения» обнаружения несущей. Существует также задержка «выключения» обнаружения несущей, возникающая при выключении несущей на другом конце линии. Цепь 109 во внутренней схеме модема необходима для фиксации цепи обмена приема данных 104 (данные принимаются только при включенном состоянии цепи 109). Задержка включения сигнала CD и фиксация цепи приема данных обеспечивают защиту от кратковременных выбросов линейных шумов, имитирующих ложные сигналы в цепи приема данных 104.

Функции цепей обмена 105 и 106.

105 (4) RTS (запрос передачи) к DCE;

106 (5) CTS (готовность к передаче) к DTE.

Сигнал RTS разрешает модулятору выход в линию (рис. 2.36). Если RTS = ВКЛ аналоговые сигналы модулятора разрешены, если RTS = ВЫКЛ выдача аналоговых сигналов в линию запрещена. В логической схеме модема RTS через схему временной задержки управляет сигналом CTS «Готовность к передаче» (в США этот сигнал называют «Свободно для передачи»). Задержка между моментами включения RTS и CTS называется «задержкой реверсирования передачи». В зависимости от типа модема она составляет от десятков до сотен миллисекунд. В период действия этой задержки разрешена передача аналоговых сигналов по линии к демодулятору на другом конце. Структура сигналов в течение этого промежутка времени зависит от типа модема: для V.21 и V.23 он состоит из сплошных единиц, отображая состояние цепи передачи данных 103 (контакт 2); для «интеллектуальных» модемов это время используется для автоматической настройки на параметры удаленного модема («протокол рукопожатия»).

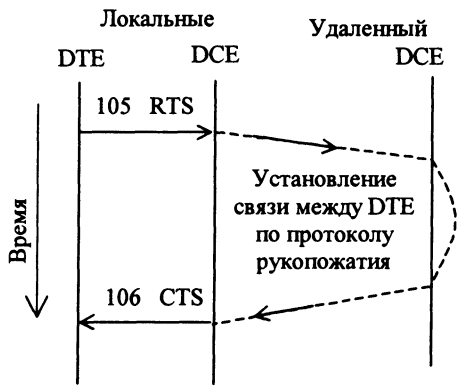


Рис. 2.36. Функции цепей обмена 105 и 106

Функции цепей обмена 107, 108, 125.

107 (6) DSR (готовность АПД) к DTE;
108/1 (20) CDSTL (подключить АПД к линии) к DCE;
108/2 (20) DTR (готовность терминала) к DCE;
125 (22) RI (индикатор вызова) к DTE.

Сигналы DSR и CDSTL/DTR используют для выполнения операции автоответа (рис. 2.37). С помощью переключателей или АТ-команд модем может быть установлен для работы в одном из режимов:

с сигналом CDSTL (подключить модем к линии);
с сигналом DTR (готовность терминала).

В режиме работы с сигналом DTR DTE включает сигнал DTR как только закончит другие операции, либо предыдущую операцию автоответа. Если модем, подключенный к DTE и находящийся в режиме автоответа, принимает входящий вызов (RI = ВКЛ), то при DTR = ВКЛ модем автоматически отвечает

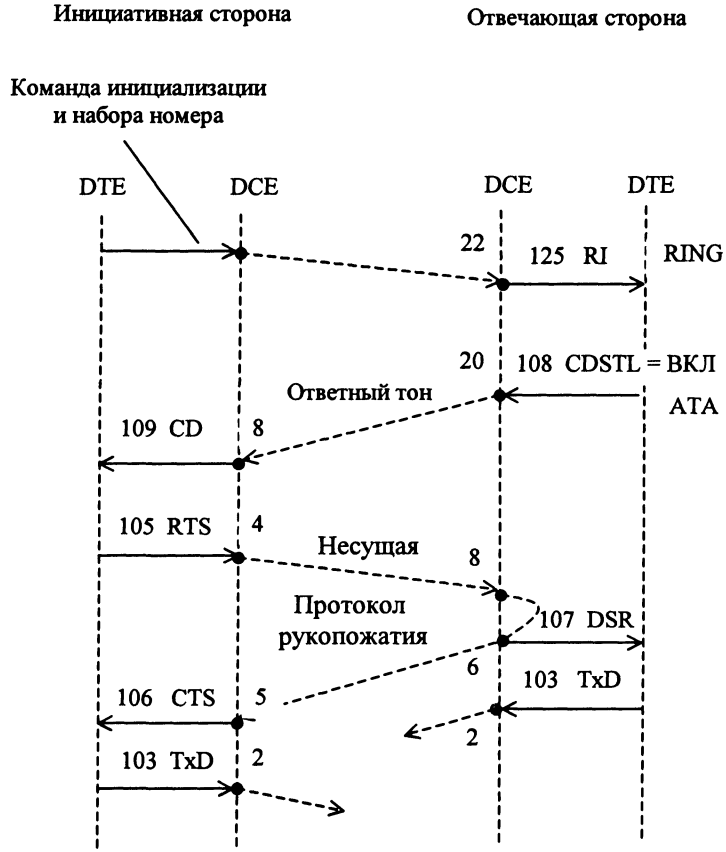


Рис. 2.37. Функции цепей обмена 125, 108, 107

на вызов и подключается к линии («Готовность АПД» (DSR) изменяет состояние из ВЫКЛ на ВКЛ). Обмен несущими между модемами вызывает изменение состояния сигнала CD («Обнаружение несущей») с ВЫКЛ на ВКЛ. После этого оба DTE могут обмениваться данными. Для разъединения канала DTE выключает сигнал DTR на 200 мс.

В режиме работы с сигналом CDSTL DTE поддерживает сигнал CDSTL в состоянии ВЫКЛ. Когда модем, подключенный к DTE, принимает вызов, он попеременно включает и выключает сигнал RI (Индикатор вызова) в такт с сигналом звонка. DTE программным способом (по состоянию регистра статуса модема) обнаруживает переход из состояния ВЫКЛ в состояние ВКЛ сигнала RI и включает сигнал CDSTL для ответа на вызов и подключения модема к линии. Далее, аналогично работе с сигналом DTR, осуществляется обмен несущими, который вызывает изменение состояния сигнала CD с ВЫКЛ на ВКЛ, после чего оба DTE могут обмениваться данными. Для разъединения DTE выключает сигнал CDSTL.

АТ-команды интеллектуального модема. Интеллектуальный модем работает в двух режимах: командном и режиме данных. В командном режиме модем интерпретирует данные, полученные с последовательного интерфейса, как «команды», и посылает обратно результаты действия как ответ. В режиме данных модем модулирует данные, полученные с последовательного интерфейса, для передачи их в линию и, наоборот, посылает демодулированные данные на последовательный интерфейс как полученные данные. Пользователю нужно знать, находится его модем в командном режиме или в режиме данных, и как переключать режимы.

Набор команд АТ – это используемый в промышленности «де факто» стандарт для управления модемом в командном режиме. Название этого набора объясняется тем, что каждая командная строка имеет префикс АТ (от слова ATtention – внимание).

Команда АТ была впервые использована в Hayes Smartmodem: она учитывается изготовителями других модемов и авторами коммуникационных программ. Исходный набор команд АТ (набор номера, ответ и т. п.) используют все, он стандартизован. Но каждый изготовитель модемов использует свой расширенный набор команд АТ для управления функциями и возможностями модема, которые он добавляет к исходному набору. Такой расширенный набор команд АТ не стандартизирован.

Набор команд АТ используют только для асинхронного последовательного интерфейса данных. Чтобы послать команду АТ на модем на компьютере, к которому он присоединен, должна быть запущена терминальная программа.

Для управления модемом АТ префикс ставится перед одной или несколькими командами одной командной строки. Префикс не ставится только перед командой A/ (повторить последнюю команду) и перед ESCAPE-последовательностью. В одной командной строке можно размещать несколько команд, но не больше, чем допускает буфер командной строки модема. Посылается команд-

ная строка на исполнение нажатием клавиши <ENTER>. Команды могут быть набраны как строчными, так и прописными буквами латинского алфавита и должны содержать численные параметры, если того требует формат команды. Если численный параметр пропущен, то его значение по умолчанию принимается равным нулю. Для редактирования неправильно набранной команды используют клавишу <BACKSPACE>.

Таблица 2.11. **Набор стандартизированных ответов**

Отклик модема	Описание
BUSY	Занято
OK	Выполнение команды
ERROR	Ошибка
NO CARRIER	Пропала несущая или соединение не установлено
NO DIALTONE	Нет длинного гудка
CONNECT	Соединение установлено

При выполнении команды модем посылает DTE ответ в виде английских слов или набора цифр. Набор стандартизованных ответов (result codes) модема представлен в табл. 2.11.

Основные регистры модема. Интеллектуальные модемы имеют три типа внутренней энергонезависимой памяти:

- постоянная память, содержимое которой устанавливается на заводе-изготовителе и доступно пользователю только на чтение;
- программируемая постоянная память, содержимое которой может меняться пользователем;
- рабочие S-регистры модема, которые определяют его текущие настройки.

Содержимое регистров можно считывать и изменять программным способом. Для чтения содержимого используется команда Sn?, где n номер регистра, для записи числа r команда Sn = r .

Перечень основных регистров модема представлен в табл. 2.12

Таблица 2.12. **Основные регистры модема**

Регистр	Предельное значение	Значение по умолчанию	Примечание
*S*0	0...255	00	Количество гудков для автоответа
*S*1	0...255	00	Счетчик приходящих гудков
*S*2	0...127 (ASCII)	43	ASCII-код ESCAPE-символа
*S*3	0...127 (ASCII)	13	ASCII-код ENTER-символа
*S*4	0...127 (ASCII)	10	ASCII-код LINEFEED
*S*5	0...32,127 (ASCII)	08	ASCII-код BACKSPACE
*S*6	2...255 (с)	02	Ожидание первого гудка, с
*S*7	1...255 (с)	45	Время одной попытки, с
*S*8	0...255 (с)	02	Задержка-запятая, с
*S*9	1...255 (0,1с)	06	Определение несущей
*S*10	1...255 (0,1с)	07	Потеря несущей
*S*11	50...255 (0,001с)	95	Скорость набора для TONE
*S*12	20...255 (0,02с)	50	Пауза для ESCAPE-последовательности

S0 – количество гудков для автоответа (*Ring to Answer On*). Содержимое этого регистра определяет количество гудков, после которого модем, находящийся в режиме автоответа, должен установить связь с удаленным модемом. S0 = 0 запрещает режим автоответа.

S1 – счетчик приходящих гудков (*Ring Count*). Эта функция работает при ненулевом значении регистра S0. Если после очередного гудка в течение 8 с следующего гудка не последует, то содержимое S1 обнуляется.

S2 – ASCII-код символа ESCAPE. По умолчанию код символа «+»(плюс). Можно заменить любым ASCII-кодом от 0 до 127.

S3 – ASCII-код символа ENTER. Символ «перевод каретки», по умолчанию 13. Можно заменить любым ASCII-кодом от 0 до127.

S4 ASCII-код символа LINEFEED. Символ «протяжка строки», по умолчанию 10. Можно заменить любым ASCII-кодом от 0 до 127.

S5 ASCII-код символа BACKSPACE. Символ «забивки», по умолчанию 8. Можно заменить любым неотображаемым ASCII-кодом от 0 до 32 или кодом 127.

S6 ожидание первого гудка (*Wait for Dial Tone*). Определяется время в секундах, в течение которого должен прийти гудок из линии при «поднятии трубки» (off-hook), по умолчанию 2 с. Если в течение этого времени придет сигнал (непрерывный гудок), то модем начнет набор номера, если нет, то модем «положит трубку» (on-hook). Содержимое регистра можно изменять в пределах от 2 до 255.

S7 время одной попытки соединения (*Wait-Time for Carrier Before Abort*). Определяет время в секундах, в течение которого должна быть установлена связь с удаленным модемом. Если в течение этого времени связь будет установлена, то модем выдаст сообщение *CONNECT*, если не будет, то модем «положит трубку» и выдаст сообщение *NO CARRIER*. По умолчанию значение зависит от типа модема (30 или 45 с). Максимальное значение регистра S7 составляет 255 с.

S8 время паузы для команды «,» (запятая), по умолчанию 2 с.

S9 время определения несущей (*Carrier Detect Response Time*). Определяет время, по истечении которого должен включаться сигнал в цепи *CD* с момента обнаружения несущей частоты от удаленного модема в линии, чтобы модем установил с ним связь. Интервал установки значения 0,1 с, по умолчанию 06 (т. е. 0,6 с).

S10 потеря несущей (*Carrier Loss to Hang Up Delay*). Определяет время, в течение которого может отсутствовать несущая частота от удаленного модема и при этом не будет оборвана связь. Интервал установки 0,1 с, по умолчанию 07 (т. е. 0,7 с). Содержимое регистра S10 должно быть всегда больше содержимого регистра S9.

S11 скорость набора для режима TONE. Определяет длительность передачи цифры и промежутка между цифрами в миллисекундах при наборе номера по методу TONE. По умолчанию значение зависит от фирмы-изготови-

теля (70, 95 и т. п.). Значение 70 примерно соответствует скорости 7,14 цифра/с. При значении 255 скорость минимальна 1,9 цифра/с. Содержимое регистра S11 не оказывает влияния на набор номера в режиме *PULSE*. Скорость передачи импульсов в режиме *PULSE* постоянна и равна 10 имп/с.

S12 пауза в ESCAPE-последовательности (*Escape Code Guard Time*). Определяет время задержки между последним *ESCAPE*-символом и следующим символом данных, переданным DTE, к которому подключен модем. Интервал установки 0,02 с, по умолчанию 50 (т. е. 1 с). Минимальное значение 20 (т. е. 0,4 с), максимальное 255 (т. е. 5,1 с).

Список основных АТ-команд. A – автоответ. Модем немедленно переводится в состояние «*off-hook*» (трубка снята), передает ответный тон и ожидает несущую от удаленного модема. Команда, введенная после *A*-команды, игнорируется.

A/ – повторение последней команды. Повторяет командную строку из буфера. Не требует АТ префикса и нажатия клавиши <*ENTER*>. Обычно используется для повторного набора номера после сигнала ЗАНЯТО.

AT – префикс командной строки. Очищает командный буфер и информирует модем о скорости передачи и формате данных.

Bn – выбор протокола Bell или CCITT. $n = 0$ – *CCITT* ; $n = 1$ – *BELL* .

D – набор номера, следующего за этой командой и установление связи в оригинальном режиме.

En – отображение на экране дисплея (эхо). $n = 0$ запрещает отображение на экране командных строк. $n = 1$ любой символ, посланный в модем в командном режиме, отображается на экране дисплея.

Hn – подключение модема к линии. $n = 0$ – отбой (on-hook). $n = 1$ – «поднять трубку» (off-hook). Модем автоматически подключается к линии, отвечая на вызов, если содержимое регистра *S*0 отлично от нуля.

In – идентификация параметров модема. $n = 0$ отображает код товара; $n = 1$ отображает контрольную сумму ПЗУ; I2 выполняет тест ПЗУ и выдает OK или ERROR.

K – листинг возможных команд. На экран дисплея выводится полный список АТ-команд, поддерживаемых модемом.

Ln – управление уровнем звука динамика. $n = 0$ – низкий; $n = 1$ – низкий; $n = 2$ – средний; $n = 3$ – высокий.

Mn – включение/отключение динамика. $n = 0$ – выключен всегда; $n = 1$ – включен только в процессе вызова или ответа; $n = 2$ – включен всегда.

O – режим передачи данных. Возврат модема в режим передачи данных, после переключения в командный режим по ESCAPE-последовательности.

P – набор номера в пульсовом (PULSE) режиме. Переводит модем в режим набора номера в пульсовом режиме. Команда *P* ставится перед номером телефона в D-команде. В пульсовом режиме каждая цифра номера передается отдельно в виде комбинации импульсов на АТС для коммутации телефонного канала.

Qn – выдача ответов на команду DTE модемом. $n = 0$ разрешает выдачу сообщений модемом (по умолчанию); $n = 1$ запрещает.

R – связь в режиме автоответа. Эту команду помещают в конце командной строки команды D. После установления связи с удаленным модемом, находящимся в оригинальном режиме, переводит ваш модем в режим автоответа. Команда *R* необходима, если удаленный модем не может работать в режиме автоответа. Режим автоответа отличается от так называемого оригинального режима тем, что при этом для приема данных используются частоты являющиеся частотами передачи для оригинального режима, а для передачи используются частоты, являющиеся частотами приема для оригинального режима.

Sn? – чтение регистра (*n* – номер регистра). Считывает содержимое регистра с номером *n* и выдает его на экран.

Sn=r – запись в регистр. Число *r* (0...255) записывается в регистр модема *n*.

T – набор номера в тональном (TONE) режиме. Переводит модем в режим набора в тональном режиме. Команда *T* ставится перед номером телефона в *D*-команде. В этом режиме номер в виде комбинации частот передается на АТС для коммутации телефонного канала.

Vn – выбор формата сообщений модема DTE. $n = 0$ – представление в виде цифрового кода; $n = 1$ – представление в виде слов (по умолчанию).

Xn – выбор набора диагностических сообщений модема. Число *n* задает набор сообщений модема DTE ($n = 0 \ldots 4$).

Z – первоначальная установка. Эта команда приводит к следующим событиям:

- модем «кладет трубку»;
- содержимое всех *Sn*-регистров возвращается к состоянию по умолчанию;
- очищается командный буфер;
- считывается состояние конфигурационных переключателей и модем устанавливается в состояние в соответствии с их конфигурацией;
- осуществляется самотестирование модема;
- посылается сообщение OK.

&Zn m – запоминание командной строки набора номера в энергонезависимой памяти. Строка *m* может содержать до 32 символов: цифры от 0 до 9, а также T, P, R, W, @, ! и ; Всего может быть запомнено 4 строки, каждой из которых ставится в соответствие определенный *n*-символ – либо 0, 1, 2, 3, либо (,), [,] для первой, второй, третьей и четвертой строки соответственно. Набор по запомненному номеру осуществляется при помощи команды **DSn**.

@ – ожидание молчания. Эта команда, помещенная между цифрами телефонного номера в команде *D*, заставляет модем ждать 30 с сигнала связи и следующего за этим сигналом 5 с «молчания», затем модем продолжает набор номера. Задержка в 30 с определяется содержимым регистра *S*7. Команда @ обычно используется при связи с компьютером, у которого стоит защита по доступу в форме требования к временным интервалам при наборе номера.

+++ – ESCAPE-символы. Появление ESCAPE-последовательности в режиме передачи данных переводит модем в командный режим. ESCAPE-последовательность состоит из первой паузы ожидания (guard time), ESCAPE-символов, второй паузы ожидания. После того как модем воспримет ESCAPE-последовательность, он выдает сообщение OK и переходит в командный режим. По умолчанию ESCAPE-символом является «+» (ASCII 43) и пауза ожидания равна 1 с. Можно изменить ESCAPE символ, изменив ASCII код в регистре S2, а также изменить продолжительность паузы ожидания, изменив содержимое регистра S12 (от 20 до 255).

! – короткая задержка при отключении связи. Эта команда имитирует процесс «положить трубку» (on-hook) на 0,5 с.

, – задержка перед набором следующей цифры, по умолчанию 2 с. Определяется содержимым регистра S8.

; – переход в командный режим работы. Если команда «;» стоит в конце командной строки команды D, то после установления связи модем переходит в командный режим работы.

W – ожидание второго длинного гудка. Команда *W* полезна для связи по междугородному номеру, когда после восьмерки или кода города необходимо дождаться длинного гудка. Для этой цели можно использовать и запятую (,), изменяя задержку в регистре S8.

Примеры применения AT-команд при наборе номера

ATD1234567<CR>

В соответствии с этой командой модем наберет в режиме, установленном по умолчанию (TONE или PULSE) телефонный номер 1234567 и будет ожидать несущую (сигнал установления связи) от удаленного модема. Если несущая не обнаружена в течение заданного интервала времени (по умолчанию 30 с), то модем автоматически разрывает линию и посылает сообщение NO CARRIER на экран DTE. Если несущая обнаружена, модем выдает сообщение CONNECT и переходит в состояние on-line, обеспечивающее взаимодействие с удаленным модемом.

ATD9,1234567<CR>

Эта команда обеспечивает выход в городскую телефонную сеть абонентов учрежденческих АТС. Для выхода необходимо набрать цифру (обычно 9) и дождаться второго гудка (команда «,»). По умолчанию одна запятая обеспечивает паузу в 2 с. Для увеличения задержки можно использовать подряд несколько запятых.

ATD1234567R<CR>

Для обеспечения связи с модемом, который может работать только в оригинальном режиме, необходимо после набора номера перевести модем в режим автоответа при помощи команды *R*.

ATD9,1234567;<CR>

Точка с запятой после номера возвращает модем после набора номера в командный режим. Это полезно, когда модем используется как автоматическое наборное устройство для последующего телефонного разговора. После ввода этой командной стоки в динамике модема прослушивается сигнал вызова. Когда вызываемый абонент снимет трубку, необходимо поднять трубку своего телефонного аппарата, а модему послать команду ATH (положить трубку). После этого можно разговаривать по телефону.

ATD1234567 @ 12345<CR>

При выполнении командной строки после набора первого номера выдерживается 5 с. Набор второго номера продолжается только в том случае, если в течение этого времени сохраняется «пауза молчания».

ATD1,234567 ! 123<CR>

В этом примере модем набирает 1, выжидает 2 с паузу, далее на 0,5 с «кладет трубку», поднимает ее и продолжает набор оставшихся цифр 123. Кратковременная операция «положить трубку» применяется в некоторых системах для передачи вызова.

ATDP&1234567<CR>

Модем автоматически в режиме *PULSE* будет повторять набор номера до установления соединения. Пауза между наборами определяется содержимым регистра S7.

AT&Z1234567<CR>

Модем сохраняет в ячейке программируемой памяти, отведенной под первую командную строку номер 1234567 и выдает сообщение OK.

ATDS<CR>

Модем осуществляет набор номера из памяти, записанного в нее в предыдущем примере.

Иногда при подключении модема к местной АТС возникает ситуация, при которой модем не различает длинного гудка при поднятии трубки и по истечении времени ожидания не осуществляет выполнение команды набора номера, а выдает сообщение NO DIALTONE.

В этом случае необходимо отключить реакцию модема на эту ситуацию, прописав в строке инициализации модема команду X1 или X3.

3. ПРИНЦИПЫ ПОСТРОЕНИЯ ЛОКАЛЬНЫХ СЕТЕЙ ЭВМ

Эта глава посвящена принципам построения и основным технологиям локальных сетей. Здесь приведена классификация локальных сетей и методов доступа к среде передачи данных, а также рассмотрены особенности применения модели OSI к локальным сетям. Дано описание практически всех основных технологий локальных сетей. Представлены основные типы оборудования и устройств локальных сетей, их функциональное назначение и характеристики. Проведен сравнительный анализ сетевых устройств.

3.1. Классификация локальных сетей и методов доступа

Локальная сеть – это коммуникационная система, поддерживающая в пределах здания или некоторой другой ограниченной территории один или несколько высокоскоростных каналов передачи цифровой информации, предоставляемых подключенным устройствам для кратковременного монопольного использования.

Классификация локальных сетей

Все многообразие существующих в мире локальных сетей можно классифицировать по следующим основным признакам: область применения, топология, методы управления доступом к среде передачи данных, программное и техническое обеспечение.

В классификации по применению необходимо выделить два основных класса локальных сетей: локальные вычислительные сети ЛВС (LCN – Lokal Computing Network) и локальные информационно-вычислительные сети ЛС (LAN – Lokal Arrea Network).

Локальные вычислительные сети призваны решать задачи рационального использования ресурсов ЭВМ, распределенных на небольшой территории, в автоматизированных системах научных исследований, системах автоматизированного проектирования и других системах, требующих доступа в реальном масштабе времени к локально-распределенным вычислительным ресурсам.

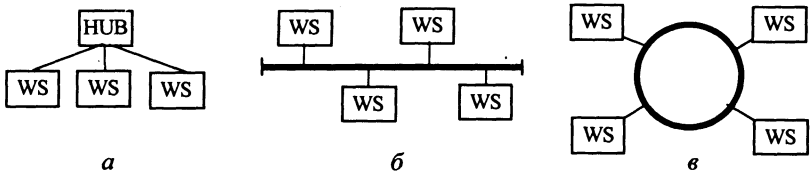


Рис. 3.1. Базовые топологии локальных сетей

Локальные информационно-вычислительные сети в основном решают задачи информационного обслуживания пользователей, реализуя функции доступа к удаленным файлам, электронной почты, передачи и обработки текстов и изображений.

По топологическим признакам локальные сети можно представить тремя базовыми топологиями: *звезда, общая шина, кольцо* (рис. 3.1). Метод конфигурации типа «звезда» является частным случаем иерархической структуры. В этой структуре центральный узел (Hub) локальной сети выполняет роль коммутатора, он производит однонаправленное соединение одной рабочей станции (WS – work station) с другой после обнаружения вызова со стороны инициатора соединения.

Основными преимуществами звездной топологии являются относительная простота их логической и программной структуры, а также возможность соединения двух абонентов на физическом уровне; а основными недостатками – низкая эффективность использования канала связи, сложность реализации в центральном узле кодового и скоростного преобразования при создании неоднородных сетей.

Наибольшая эффективность использования канала связи достигается при конфигурации «общая шина», в которой значительно упрощаются логическая и программная структура локальной сети. Топология «общая шина» обеспечивает простоту расширения сети посредством добавления дополнительного числа рабочих станций, подключаемых к магистрали, сохраняя простоту методов управления, возможность работы в параллельном коде. Данная конфигурация позволяет осуществлять широковещательную передачу, когда информация, передаваемая одним абонентом, воспринимается всеми, подключенными к сети абонентами, одновременно.

Недостатком такой конфигурации является последовательный характер использования магистрали (по «общей шине» может передаваться в определенный момент времени только один кадр).

К структурам со слабой централизацией управления наряду с конфигурацией «общая шина» относится и кольцевая структура, обеспечивающая контроль работоспособности канала посылкой по кольцу «эхо»-сигнала, и возможность использования однонаправленных усилителей сигналов, т. е. применять волоконно-оптические линии связи (ВОЛС).

Классификация локальных сетей по методам управления доступом к среде передачи данных приведена на рис. 3.2. С развитием локальных сетей будут развиваться и методы доступа, поэтому приводимый перечень не полный.

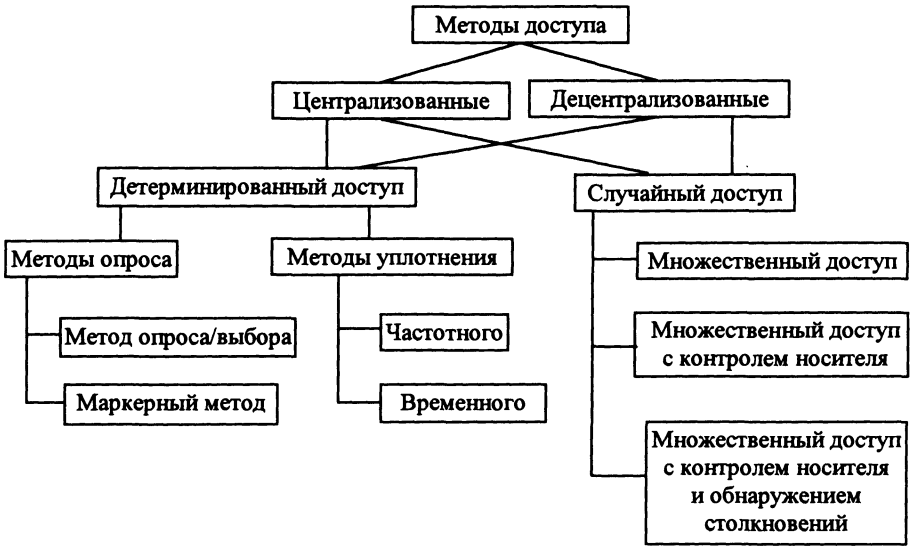


Рис. 3.2. Классификация методов доступа

Все сети независимо от топологии и реализуемого метода доступа к среде передачи данных имеют некоторые общие компоненты, функции и характеристики. В их числе:

- серверы (server) – компьютеры, предоставляющие свои ресурсы сетевым пользователям;
- среда (media) – способ соединения компьютеров;
- совместно используемые данные – файлы, предоставляемые серверами по сети;
- совместно используемые периферийные устройства (принтеры, библиотеки CD-ROM и т. д.) – ресурсы, предоставляемые серверами;
- ресурсы – файлы, принтеры и другие элементы, используемые в сети.

По принципу предоставления ресурсов в совместное использование сети разделяются на три типа: *одноранговые* (peer-to-peer), *на основе сервера* (server based) и *комбинированные*. Различия между ними имеют принципиальное значение, поскольку определяют разные возможности этих типов сетей. Выбор типа сети зависит от многих факторов: размера предприятия; необходимого уровня безопасности; уровня доступности административной поддержки; объема сетевого трафика; потребностей сетевых пользователей, финансовых затрат.

Одноранговые сети. В одноранговой сети все компьютеры равноправны: нет иерархии среди компьютеров и нет выделенного (dedicated) сервера. Как правило, каждый компьютер функционирует и как клиент, и как сервер; иначе говоря, нет отдельного компьютера, ответственного за администрирование всей сети. Все пользователи самостоятельно решают, какие данные на своем ком-

пьютере сделать общедоступными по сети. Одноранговые сети называют также рабочими группами. Рабочая группа – это небольшой коллектив, поэтому в одноранговых сетях бывает чаще всего не более 10 компьютеров.

Одноранговые сети относительно просты. Поскольку каждый компьютер является одновременно и клиентом, и сервером, нет необходимости в мощном центральном сервере или в других компонентах, обязательных для более сложных сетей. Одноранговые сети обычно дешевле сетей на основе сервера, но требуют более мощных (и более дорогих) компьютеров.

В одноранговой сети требования к производительности и к уровню защиты для сетевого программного обеспечения, как правило, ниже, чем в сетях с выделенным сервером. Выделенные серверы функционируют исключительно в качестве серверов, но не клиентов или рабочих станций (*workstation*).

В такие операционные системы, как Microsoft Windows NT Workstation, Microsoft Windows for Workgroups и Microsoft Windows 95, встроена поддержка одноранговых сетей. Поэтому, чтобы установить одноранговую сеть, дополнительного программного обеспечения не нужно.

Одноранговая сеть характеризуется рядом стандартных решений:

- компьютеры расположены на рабочих столах пользователей;
- пользователи сами выступают в роли администраторов и обеспечивают защиту информации;
- для объединения компьютеров в сеть применяется простая кабельная система.

Одноранговую сеть удобно использовать в коллективах до 10 человек, где пользователи расположены компактно, вопросы зашиты данных не критичны, кроме того, в обозримом будущем не ожидается значительного расширения фирмы и, следовательно, сети.

Несмотря на то, что однорангоные сети вполне удовлетворяют потребностям небольших организаций и предприятий, при решении вопроса об их использовании необходимо учитывать, что:

- в типичной одноранговой сети системный администратор, контролирующий всю сеть, не выделяется, и каждый пользователь сам администрирует свой компьютер (сетевое администрирование (administration) решает ряд задач, в том числе: управление работой пользователей и защитой данных; обеспечение доступа к ресурсам; поддержка приложений и данных; установка и модернизация прикладного программного обеспечения);
- все пользователи могут «поделиться» своими ресурсами с другими (к совместно используемым ресурсам относятся каталоги, принтеры, факс-модемы и т. п.);
- в одноранговой сети каждый компьютер должен большую часть своих вычислительных ресурсов предоставлять локальному пользователю, а для поддержки доступа к ресурсам удаленного пользователя (обращающегося к серверу по сети) подключать дополнительные вычислительные ресурсы (сеть на основе сервера требует более мощных серверов, поскольку они должны обрабатывать запросы всех клиентов сети);

• централизованно управлять защитой (защита подразумевает установку пароля на разделяемый ресурс, например, на каталог, файл) в одноранговой сети очень сложно, так как каждый пользователь устанавливает ее самостоятельно (такая ситуация представляет серьезную угрозу для всей сети, кроме того, некоторые пользователи могут вообще не установить защиту);

• поскольку в одноранговой сети каждый компьютер функционирует и как клиент, и как сервер, пользователи должны обладать достаточным уровнем знаний, чтобы работать и как пользователи, и как администраторы своего компьютера.

Сети на основе сервера. *Выделенным* называется такой сервер, который функционирует только как сервер (исключая функции клиента или рабочей станции). Он специально оптимизирован для быстрой обработки запросов от сетевых клиентов и для управления защитой файлов и каталогов. Сети на основе сервера являются промышленным стандартом.

В сети может быть подключено несколько серверов. Круг задач, которые должны выполнять серверы, многообразен и сложен. Чтобы приспособиться к возрастающим потребностям пользователей, серверы в больших сетях стали *специализированными* (specialized). Распределение задач среди специализированных серверов гарантирует, что каждая задача будет выполняться самым эффективным способом из всех возможных. Например, в сети Windows NT работают различные типы серверов:

• *файл-серверы и принт-серверы*, управляющие доступом пользователей соответственно к файлам и принтерам. Например, чтобы работать с текстовым процессором, вы прежде всего должны запустить его на своем компьютере. Документ текстового процессора, хранящийся на файл-сервере, загружается в память вашего компьютера, и таким образом вы можете работать с этим документом на своем компьютере. Другими словами, файл-сервер предназначен для хранения файлов и данных;

• *серверы приложений*, выполняющие прикладные части клиент-серверных приложений, в них также находятся данные, доступные клиентам. Например, чтобы упростить получение данных, серверы хранят большие объемы информации в структурированном виде. Эти серверы отличаются от файл- и принт-серверов. В последних на запрашивающий компьютер пересылают только результаты запроса. Приложение-клиент на удаленном компьютере получает доступ к данным на сервере приложений. Однако вместо всей базы данных на удаленный компьютер с сервера загружаются только результаты запроса в виде экранных форм;

• *почтовые серверы*, управляющие передачей электронных сообщений между пользователями сети;

• *факс-серверы*, управляющие потоком входящих и исходящих факсимильных сообщений через один или несколько факс-модемов;

• *коммуникационные серверы*, управляющие потоком данных и почтовых сообщений между вашей сетью и другими сетями или удаленными пользователями через модем и телефонную линию.

В локальных сетях на основе выделенного сервера существует специальная служба, предназначенная для поиска, хранения и защиты информации в сети. Например, Windows NT Server объединяет компьютеры в логические группы – домены (domain), система защиты которых наделяет пользователей различными правами доступа к любому сетевому ресурсу.

Таким образом, можно выделить ряд преимуществ, которыми обладают локальные сети с выделенным сервером:

• *разделение ресурсов.* Сервер проектируют таким образом, чтобы предоставить доступ к множеству файлов и принтеров, обеспечивая при этом высокую производительность и защиту. Администрирование и управление доступом к данным осуществляется централизованно. Ресурсы, как правило, расположены также централизованно, что облегчает поиск и поддержку;

• *защита данных.* Основным аргументом при выборе сети на основе сервера является, как правило, защита данных. В таких сетях, например, как NetWare Novell или Windows NT Server, проблемами безопасности может заниматься один администратор: он формирует политику безопасности (security policy) и применяет ее в отношении каждого пользователя сети;

• *резервное копирование данных.* Поскольку жизненно важная информация расположена централизованно, т.е. сосредоточена на одном или нескольких серверах, нетрудно обеспечить ее регулярное резервное копирование (backup);

• *избыточность.* Благодаря избыточным системам данные на любом сервере можно дублировать в реальном времени, поэтому в случае повреждения основной области хранения данных информация не будет потеряна – легко воспользоваться резервной копией;

• *количество пользователей.* Сети на основе сервера способны поддерживать тысячи пользователей. Сетями такого размера, будь они одноранговыми, было бы невозможно управлять;

• *аппаратное обеспечение.* Так как компьютер пользователя не выполняет функций сервера, требования к его характеристикам зависят от потребностей самого пользователя.

Комбинированные сети. Такие сети совмещают лучшие качества одноранговых сетей и сетей на основе сервера.

Многие администраторы считают, что подобная сеть наиболее полно удовлетворяет запросы, так как в ней могут функционировать оба типа операционных систем. Операционные системы для сетей на основе сервера, например Microsoft Windows NT Server или Novell® NetWare®, в этом случае отвечают за совместное использование основных приложений и данных. На компьютерах-клиентах могут выполняться операционные системы Microsoft Windows NT Workstation или Windows 95, которые будут управлять доступом к ресурсам выделенного сервера и в то же время предоставлять в совместное использование свои жесткие диски, а по мере необходимости разрешать доступ и к своим данным.

Особенности применения эталонной модели взаимодействия открытых систем к локальным сетям

Для применения эталонной модели МОС в задачах локальных сетей комитет по локальным сетям в проекте IEEE 802 выполнил дальнейшую декомпозицию физического и канального уровней (рис. 3.3). Канальный уровень делится на два подуровня: *управление логическим звеном* УЛЗ (LLC – Logical Link Control) и *управление доступом к передающей среде* УДС (MAC – Medium Access Control).

В функции подуровня УЛЗ входит передача кадров между станциями, установление логического соединения, контроль ошибок. Подуровень УДС реализует алгоритм доступа к среде и адресацию станций.

Физический уровень делится на три подуровня: *физической сигнализации* ФС (PS – Physical Signalling), *интерфейса с устройством доступа* (AUI – Access Unit Interface), *подключения к физической среде* (PMA – Physical Medium Attachment). Подуровень физической сигнализации выделяется для облегчения схемной интеграции с канальным уровнем и выполняет функции кодирования и передачи двоичных символов, их приема и декодирования. Средства подключения к физической среде согласуют сигналы из уровня передачи физических сигналов с требованиями передающей среды. Интерфейс с устройством доступа представляет собой кабель и позволяет размещать подключаемую станцию локальной сети на некотором расстоянии от физического носителя информации.

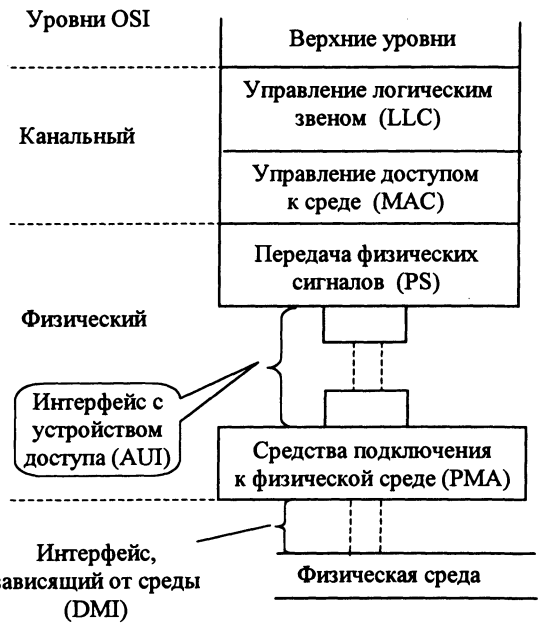


Рис. 3.3. Декомпозиция уровней модели OSI ISO для локальных сетей

Рассмотрим последовательность действий по преобразованию информации в форму, пригодную для транспортировки по моноканалу локальной сети.

Сборка пакета осуществляется последовательно под управлением протоколов процессов пользователя, сеанса, передачи и пересылки пакетов. Обычно пакет формируется в среде ЭВМ, которая помимо этого осуществляет обработку прикладных программ пользователей. Для локальной сети протокол пересылки пакетов не содержит средств оптимизации маршрутов пакетов и используется в полную силу лишь в машинах-шлюзах (маршрутизаторах).

Характеристики, относящиеся к этапам формирования пакетов, представляют интерес, в основном, для глобальных сетей, поэтому они рассмотрены в соответствующем разделе учебного пособия. Специфичными же для локальных сетей являются характеристики, определяющие процессы сборки кадра и его транспортировки на канальном и физическом уровнях модели ВОС.

Формирование кадра из пакета информации осуществляется в адаптере локальной сети источника, реализующего два базовых протокола управления каналом УЛЗ (LLC) и УДС (MAC). Сформированный кадр передается по моноканалу в приемник, где подвергается разборке в адаптере локальной сети приемника. Выделенный из кадра пакет передается в зону действия ЭВМ станции локальной сети, в которой при помощи протоколов высокого уровня из пакета выделяется информация пользователя.

Методы доступа к среде передачи данных

Разделяемым ресурсом в локальных сетях является пропускная способность канала передачи данных. Существуют различные способы разделения пропускной способности канала, основанные на методах детерминированного доступа и методах случайного доступа (см. рис. 3.2).

В зависимости от способа предоставления права на пользование каналом методы первой группы способов бывают централизованными и децентрализованными. В первом случае санкцию на пользование каналом в соответствии с запросами дает центральный узел, во втором каждый узел сам определяет право на пользование каналом.

Аналогично и методы второй группы способов подразделяют на централизованные и децентрализованные. В первом случае центральный узел управления выполняет исключительно функцию управления коммутацией и в соответствии с его указаниями остальные узлы передают либо принимают информацию из канала. Во втором случае каждый узел индивидуально оценивает ситуацию управления, передает или принимает информацию, направленную по его адресу. Этот метод широко используют в системах с коммутацией пакетов.

Методы опроса. Разделение пропускной способности каналов на основе опроса является самым простым методом при коммутации пакетов. При этом первичный (управляющий) узел периодически опрашивает по списку вторичные (подчиненные) узлы последовательно один за другим на предмет наличия запроса на передачу. Право на пользование каналом предоставляется узлу, у которого имеется такой запрос. После окончания сеанса связи первичный узел начинает циклический опрос остальных узлов из списка о наличии запроса на

передачу. Описанный метод наиболее часто используют в недорогих системах с невысоким быстродействием, имеющих звездообразную структуру или общую шину. Примерами стандартов, регламентирующих протоколы для таких систем, являются BSC и IBM 2848.

Рассмотрим подробнее один из способов реализации метода опроса на примере системы с *опросом/выбором*. Назначение команды *Опрос* (Polling) состоит в передаче данных первичному узлу. Назначение комады *Выбор* (Selecting) противоположное – передавать данные из первичного узла во вторичный. Эти команды являются основными для передачи данных в любой узел канала или сети. Прежде всего команда Опрос посылается из первичного узла во вторичный узел. Если во вторичном узле имеются данные, готовые к передаче, они посылаются в опрашивающий узел. Первичный узел осуществляет контроль ошибок и посылает положительный ответ АСК (при отсутствии ошибок) и отрицательный ответ NAC (при обнаружении ошибки). Эти два события (передача данных и АСК/NAC) могут происходить много раз до тех пор, пока у вторичного узла не останется данных, которые необходимо передать. В этом случае вторичная станция должна послать уведомление EOT (End Of Transmission) о том, что она закончила передачу.

Посылка команды Выбор вторичному узлу фактически означаетпроверку готовности вторичного узла принять данные от первичного. Подтверждение АСК в ответ на выбор означает положительный ответ. Данные передаются, проверяются на наличие ошибок, и их прием подтверждается. Процесс повторяется до окончания данных. В конце сеанса передачи посылается управляющее уведомление EOT. В более совершенных системах при установлении связи в узлах резервируются ресурсы для приема данных. Поэтому считается, что приемник в любой момент может получить данные, и необходимость в командах выбора отпадает.

Иногда на команду выбора вторичный узел отвечает отрицательным подтверждением NAC. Существует множество причин, почему узел не может вести прием данных. Он может быть занят выполнением других задач или не иметь необходимого пространства в буфере для приема данных; у него могут иметься данные для передачи в первичный узел. Система опроса/выбора решает проблему таким образом, что первичный узел инициирует опрос, который позволяет вторичному узлу послать данные и очистить свои буферы.

Система опроса/выбора может быть использована в режиме соединения «точка-многоточка» для разделения пропускной способности канала между двумя вторичными станциями. Например, устройство В хочет обменяться данными с устройством А. Чтобы это произошло, необходимо, чтобы первичный узел опросил В. Для чего данные посылаются непосредственно не в А, а в первичный узел. В первичном узле данные проверяются на наличие ошибок, и посылается подтверждение АСК. Завершается процесс посылкой EOT от устройства В. После этого данные, поступившие в первичный узел, транслируются по тому же каналу в узел А. Этот процесс осуществляется процедурой выбора узла А.

Недостатком системы с опросом/выбором является наличие неоднократных ответных реакций на опрос, что приводит к непроизводительному потреблению дорогостоящих ресурсов канала. Эти накладные расходы особенно ощутимы в системах без мультиплексоров или групповых контроллеров терминального оборудования. Подобные устройства могут принять команду опроса, общую для всех устройств, произвести сканирование подсоединенных устройств для выявления активного запроса на передачу и передать данные первичному узлу.

Другой подход к уменьшению накладных расходов, вызываемых опросом, состоит в использовании динамических таблиц опроса/выбора. Если продолжается опрос терминала и он не отвечает после определенного числа попыток, его приоритет в таблице опроса понижается, и, следовательно, он реже обслуживается. Те устройства, которые положительно реагировали на опрос, продвигаются вверх в таблице приоритетов. Можно предусмотреть многократный просмотр таблицы для одного и того же устройства, т.е опросить станцию А, затем станцию С, потом снова А, так как А была ранее занята и теперь положительно отреагировала на команды опроса. Динамический опрос/выбор устраняет некоторые накладные расходы, характерные для обычных статических систем с опросом/выбором.

В системах с большим числом узлов и небольшой частотой появления запросов на передачу время циклического опроса узлов о наличии запросов является значительным. Для сокращения времени циклического опроса используют *метод зондирования*. В этом случае все узлы, кроме первичного, разделяются на группы, каждая из которых опрашивается на предмет наличия запросов на передачу. Если внутри группы есть такие запросы, то выдается положительный ответ, который посылается на управляющий узел. Для той группы узлов, от которой получен положительный ответ, управляющий узел проводит зондирование ее узлов, используя метод опроса по списку.

Одной из модификаций метода опроса узлов по списку является *метод гнездового опроса*. В этом случае все узлы, кроме управляющего, предварительно переписываются в определенном порядке. Управляющий узел опрашивает узлы о наличии запросов. При отсутствии запроса опрошенный узел опрашивает следующий по порядку узел о наличии запроса на передачу. Подобная процедура повторяется до тех пор, пока не встретится узел, содержащий запрос. При наличии запроса метод идентичен методу опроса по списку. Такая модификация фактически эквивалентна методу с передачей маркера в системах с децентрализованными запросами каналов, за исключением того, что в данном случае связь устанавливается между управляющим и другими узлами. Метод гнездового опроса позволяет сократить время опроса и, кроме того, уменьшить нагрузку управляющего узла.

Как видно из рассмотренных примеров системы, реализующие методы опроса имеют ярко выраженную иерархическую топологию: весь трафик поступает в первичный узел и выходит из него. Иерархическая топология связа-

на с потенциальной опасностью перегрузки, так как управление трафиком осуществляется одним устройством. С этой конфигурацией связаны также некоторые проблемы надежности: выход из строя первичного узла приводит к отказу всей сети.

Система с передачей маркера обычно реализуется в локальной сети с кольцевой топологией или с общей шиной. Считается, что имеет место циклическая нумерация узлов, т. е. за узлом с последним номером следует узел с первым номером. При включении в системе генерируется *маркер* – специальная кодовая последовательность, которая передается по кольцу. Все узлы, соединенные по кольцевой схеме, имеют приемный и передающий регистры, устройство для сравнения кодовых комбинаций и переключатели. Если узел имеет данные для передачи, он вынужден ждать, пока предшествующий узел не вышлет ему маркер. Когда узел получает маркер, он на время удаляет его из кольца и помещает вслед за пакетом данных, сохраняемым в сдвиговом регистре для передачи. Затем сдвиговый передающий регистр последовательно включается в кольцо, а его содержимое, включая маркер в конце пакета, посылается по кольцу. Далее регистр отключается от кольца, а узел ожидает возвращения отправленного им пакета. Первый же пакет, полученный на приемной стороне, должен при нормальных условиях быть этим отправленным пакетом. Поэтому первый же полученный пакет считывается в приемный регистр для анализа. После этого восстанавливается обычная цепь кольца, а узел переходит в состояние ожидания следующего маркера для передачи следующей порции информации. Таким образом, поступающий в некоторый узел поток информации всегда начинается пакетом, отправленным данным узлом. Каждый отправитель ответственен за удаление своих пакетов из кольца. Каждый передаваемый пакет всегда становится последним в последовательности пакетов, предшествующих маркеру. Устройство-получатель читает пакет по мере его прохождения через узел и может выставить флаг подтверждения в его конце, не изменяя при этом сам пакет.

Основные трудности в кольце с передачей маркера возникают в случае потери маркера либо неудаления своего пакета отправителем. Первая ситуация может возникнуть когда маркер удален каким-либо узлом передающим информацию, а затем не восстановлен по причине аппаратного сбоя или когда маркер оказался поврежденным при передаче и стал поэтому нераспознаваем. Пакет может оказаться неудаленным потому, что произошла ошибка в узле-отправителе, и поток поступающей информации не был отведен в приемный буфер этого узла. С обеими ситуациями удается справиться с помощью специального устройства, ответственного за наличие маркера и следящего за циркуляцией пакетов по кольцу, или, поручив выполнять эти функции одному из узлов.

Возможно дублирование маркера, если какие-то два узла генерируют новые маркеры одновременно. Этого можно избежать, если каждый узел, генерирующий маркер, всегда помещает перед маркером полный пакет и следит за тем, чтобы он вернулся первым. Каждый появившийся на входе пакет прове-

ряется и удаляется, если он отличается от переданного пакета. Если два узла делают это одновременно, то они уничтожают маркеры и пакеты друг друга. После произвольного временно́го интервала в какой-то точке кольца снова генерируется маркер. Если каждый из узлов, уже передавших пакет, всегда уничтожает первые пакеты, поступившие к нему из кольца, пока не дойдет до своего пакета, то тем самым решается и проблема неудаленных пакетов.

Для систем с передачей маркера, имеющих кольцевую топологию, возможны различные варианты практической реализации: например, имеются системы, в которых удаление пакета из сети осуществляет приемный узел; системы, в которых пакет посылается вместо удаленного маркера, а восстановление маркера осуществляется после удаления своего пакета, и другие.

Принцип работы системы с передачей маркера с общей шиной такой же, как и в случае кольцевой топологии. В каждом узле для маркера устанавливается очередность прохождения остальных узлов, т. е. заранее формируется предполагаемый циклический маршрут маркера. Узлу, которому по его адресу, указанному в маркерном пакете, был передан маркер, предоставляется право пересылки информационного пакета. При отсутствии пакетов, ожидающих передачи, маркерный пакет немедленно высылается следующему по очереди узлу.

Преимуществом системы с передачей маркера с общей шиной является то, что она естественным путем обеспечивает удаление пакетов из сети и автономно определяет очередность передачи пакетов узлами и физическое местоположение маркера на шине.

К недостаткам таких систем можно отнести затраты времени на перемещение маркера, предоставляющего право на пользование шиной.

Методы уплотнения. Различают временно́е и частотное уплотнение каналов. *Временное уплотнение* реализуется при помощи мультиплексора, предоставляющего каждому из подключенных к общему каналу низкоскоростных устройств один временной такт, в течение которого это устройство получает в свое монопольное пользование быстродействующий канал, обслуживающий всю совокупность таких устройств. Сами устройства выдают данные со свойственным им быстродействием, поэтому используются короткие временны́е такты и каждое из устройств может часто обращаться к общему каналу. На противоположном конце канала необходим демультиплексор для идентификации данных, поступающих в потоке, и распределении их по отдельным устройствам. Применение метода временного уплотнения, при котором используются фиксированные временны́е такты, достаточно эффективно, если каждое из устройств постоянно передает или принимает информацию. Подобная ситуация на практике встречается исключительно редко.

Принцип, на котором основывается метод *статистического временно́го уплотнения* заключается в том, что временны́е такты предоставляются устройству лишь тогда, когда оно в них действительно нуждается. Устройства, подключенные к такому мультиплексору, могут соперничать друг с другом за

право доступа к общему каналу. Однако маловероятно, чтобы все они одновременно требовали использования разделяемого канала. Поэтому при одинаковой пропускной способности статистический мультиплексор сможет поддержать большее количество устройств, чем обычный мультиплексор с временны́м уплотнением. Статистический мультиплексор, очевидно, должен быть в достаточной степени «интеллектуальным» для того, чтобы определить, какому из устройств требуется такт, а также для того, чтобы выполнить обычную функцию уплотнения в том случае, когда нескольким устройствам одновременно требуется общий канал.

При использовании *частотного уплотнения* широкая полоса пропускания некоторой среды передачи данных разделяется на некоторое число индивидуальных каналов. Известно, что любой канал, используемый для передачи информации, имеет определенную полосу пропускания частот. В телефонной сети она очень узка. Для коаксиального кабеля полоса пропускания равна нескольким сотням мегагерц. Если имеется канал с широкой полосой пропускания, то такую полосу можно делить на несколько более узких полос, каждая из которых должна быть адекватной для выбранной скорости передачи цифровых данных. Между частотными полосами необходимы узкие защитные полосы, чтобы минимизировать интерференцию между соседними рабочими частотами. Частотное уплотнение предоставляет возможность установления связи между несколькими устройствами в некоторой полосе частот вне зависимости от взаимодействий в других диапазонах частот.

Рассмотренные методы уплотнения успешно применяют в локальных сетях. Кроме того, для локальных сетей разработаны новые разновидности этих методов для различных передающих сред, топологических структур и режимов работы. В локальных сетях редко применяется центральный мультиплексор с разделением времени или концентратор для распределения пропускной способности сети. Вместо этого ответственность за распределение пропускной способности сети обычно распределяется между всеми станциями, имеющими доступ к сети. Рабочие частоты назначаются супервизором сети, а в более сложных системах специальным управляющим устройством.

В локальных сетях с коммутацией каналов с частотным разделением используют коаксиальные кабели. По ним можно пересылать сигналы в широком частотном диапазоне. Принцип работы таких систем заключается в следующем. Каждый узел подключается к кабелю через управляемый цифровой модем. Узел, которому требуется выделить канал, посылает соответствующий запрос управляющему узлу на частотах, предназначенных для управляющих сообщений. В ответ на этот запрос управляющий узел резервирует свободную частоту для организации сеанса связи и сообщает ее передающему и приемному узлам. Абоненты настраивают соответствующие модемы на зарезервированную частоту и устанавливают связь.

При помощи коаксиальных кабелей, предназначенных для передачи сигналов в широком частотном диапазоне, можно формировать сети с передачей по

одному коаксиальному кабелю, резервируя для передачи и приема группы частот. Такие сети имеют древовидную топологию, а суммарная протяженность кабелей может достигать нескольких десятков километров с подключением по мере необходимости широкополосных усилителей.

К практически реализованным системам с временным уплотнением, в которых приоритет на пользование каналом назначается заранее, относятся системы *с переменным распределением каналов*. В этих системах право на пользование каналом уступается следующему узлу при условии, что временной такт, выделяемый для передачи пакета, в данном узле не используется. Примером данных систем являются системы с тактированным доступом и с вставкой регистра.

Системы, использующие *метод вставки регистра*, применяют в локальных сетях с кольцевой топологией. Принцип их работы следующий. Когда узел имеет информацию для передачи, он помещает ее в сдвиговой регистр. Этот регистр может быть последовательно включен (вставлен) в канал, обеспечивая передачу как собственной информации, так и транзитной. Регистр остается включенным в кольцо до тех пор, пока в него полностью не загрузится переданный ранее этим узлом пакет. Узел-получатель пакета должен прочитать данные и вставить признак того, что данные приняты. Ответственность за удаление пакета из кольца несет узел-источник информации.

Системы, использующие *метод с тактируемым доступом*, реализуют в локальных сетях с кольцевой топологией. Для них не нужны сдвиговые регистры и высокоскоростные переключатели в повторителях или подключаемых к кольцу узлах. Здесь используется один или несколько контейнерных пакетов, или тактов, непрерывно циркулирующих по кольцу. Их число никогда не меняется и определяется длиной такта, общей длиной кольца и процедурой начального запуска кольца. Если кольцо очень короткое, то короткими должны быть и используемые такты, а их число невелико, иначе придется вставлять в кольцо буфер с задержкой, так как начало такта может возвратиться к отправителю раньше, чем тот завершит передачу данного пакета. По этой причине во многих практических реализациях кольцевых сетей с тактируемым доступом применяется только один короткий такт и буфер с задержкой. В момент запуска кольца один из повторителей или узлов формирует пакет-контейнер и отправляет его по кольцу. Если он вернется к отправителю, то это будет означать, что кольцо замкнуто, и можно начинать работу.

Если у узла есть информация для передачи, то он загружает ее в буфер и ожидает, когда к нему поступит пустой контейнер. Пустой пакет-контейнер легко опознается по контрольному полю в его заголовке. При поступлении контейнера узел сдвигает пакет данных из своего буфера в поля данных пакета-контейнера по мере прохождения последнего через узел. При этом признаку, указывающему состояние такта, сначала присваивается значение «занято» и в заголовок помещается адрес назначения. Пакет-контейнер затем продолжает передаваться вдоль кольца до тех пор, пока он не достигнет узла назначения, повторитель

которого копирует содержащуюся в контейнере информацию в свой буфер и выставляет в конце такта признак, означающий, что пакет получен. Далее пакет-контейнер с признаком занятости продолжает следовать от повторителя к повторителю, пока не достигнет узла-источника. Отправитель информации опознает отправленный им пакет и переводит признак занятости контейнера в состояние «свободно», позволяя тем самым остальным узлам использовать пакет-контейнер для передачи данных. Узел-источник информации проверяет также содержимое поля подтверждения в пакете-контейнере, чтобы убедиться в том, что узел назначения действительно получил отправленный ему пакет.

Таким образом, несмотря на явные потери времени из-за того, что заполненный пакет-контейнер вынужден совершать полный оборот, он используется как для передачи данных в прямом направлении, так и для доставки подтверждения на обратном пути.

Если занятый пакет-контейнер не был освобожден узлом-источником информации (например, из-за сбоя в этом узле после передачи), то контейнер с меткой «занято» будет циркулировать по кольцу. На практике за ошибками в сети обычно следит специальное устройство, которое освобождает пакет-контейнер, проходящий мимо данного устройства в неизменном состоянии более одного раза, а также отвечает за запуск сети в работу.

Методы случайного доступа. Если нескольким узлам разрешить одновременно пересылать пакеты, то может произойти их столкновение, в результате которого информация будет испорчена. В системах случайного доступа необходимо как можно быстрее удалить поврежденные при столкновениях пакеты и освободить канал для последующих передач пакетов. Наиболее просто это реализуется в структурах с общей шиной, где удаление пакетов происходит автоматически за короткий промежуток времени. Поэтому локальные сети, в которых реализованы методы случайного доступа, имеют логическую структуру «общая шина».

Простейшей системой случайного доступа, осуществляющей *множественный доступ* к среде передачи, является локальная сеть ALOHA. Она была разработана в начале 70-х годов для обеспечения связи центральной ЭВМ Гавайского университета с терминалами, расположенными на всех островах архипелага. В этой системе использованы два канала: один отведен для передачи сообщений от ЭВМ к терминалам, другой – от терминалов к ЭВМ. В первом канале используется только одно передающее устройство, поэтому никаких трудностей с распределением канала не возникает, второй же канал используется всеми терминалами.

Если у некоторого терминала имеется пакет, готовый к отправке, терминал передает этот пакет, не обращая внимания на то, занят канал в данный момент или нет. По завершению передачи пакета терминал запускает таймер. Если по истечении определенного времени терминал не получил подтверждения от центральной ЭВМ о приеме пакета, то считается, что произошло столкновение, и терминал повторяет передачу того же пакета. Для уменьшения вероятности повторного конфликта между теми же пакетами интервал, через который терминал повторит передачу пакета, задается случайным образом.

Приемник на центральной ЭВМ принимает как нормальные, так и искаженные пакеты. Каждый пакет проверяется на наличие ошибок. Если в пакете ошибок не обнаружено, то по каналу ЭВМ – терминал, для которого конфликтная ситуация, вызываемая столкновением пакетов, исключена, посылается подтверждение о получении. Если обнаруживается ошибка, то подтверждение не посылается.

Даже если длительность временно́го промежутка, в течение которого происходит наложение пакетов, очень мала, оба пакета искажаются и их необходимо передавать заново.

Суммарная продолжительность потерянного при передаче времени исчисляется от начала передачи первого пакета до завершения передачи второго. Преимущество такой системы состоит в простоте ее реализации, а недостаток – в очень низком коэффициенте использования тракта передачи (не более 19 %) при большой нагрузке на сеть.

Одним из способов повышения производительности сети является тактирование. Центральная ЭВМ формирует серию последовательных временны́х тактов (слот-тайм), и передача пакета осуществляется только в начале каждого такта. Следовательно, конфликт может возникнуть лишь в начальной фазе такта. Подобный прием позволяет почти удвоить коэффициент использования тракта (до 37 %).

Другой способ уменьшения вероятности столкновения пакетов реализован в системе *множественного доступа с контролем носителя* (МДКН), в которой посылка пакета начинается только после освобождения среды передачи (носителя информации). Столкновения в системе МДКН возможны лишь в случае, когда два или более узла одновременно пытаются переслать пакет сразу после освобождения канала. Поэтому существуют различные способы начала передачи пакета. В соответствии с этими способами, системы МДКН подразделяются на системы *I*-, *N*- и *p*-типа.

В системах *I*-типа передача пакета начинается сразу же после освобождения тракта передачи. Вероятность возникновения столкновений в такой системе больше, чем для систем *p*- и *N*-типа.

В системах *N*-типа, если канал оказывается занятым, передача пакета откладывается на более поздний момент, чем освобождение тракта передачи, и с учетом этого осуществляется корректировка расписания пересылки пакетов. Вероятность возникновения столкновений в такой системе незначительна, однако существенно возрастает вероятность простоя канала, а коэффициент использования тракта передачи остается в целом невысоким.

Система *p*-типа представляет собой некий компромиссный вариант систем *I*-типа и *N*-типа. В этой системе после освобождения носителя посылка пакета начинается с вероятностью *p*, поэтому такой метод еще называют *p*-настойчивым МДКН. Если известна зависимость между вероятностью появления запроса на передачу пакета и длительностью передачи, то можно определить оптимальное значение вероятности *p*. Использование оптимальной величины *p*

обеспечивает небольшие вероятности возникновения столкновений и простоя тракта передачи.

Теоретические верхние границы коэффициента использования тракта передачи для систем *I*-, *p*- и *N*-типа соответственно составляют: 52, 83, 81 %.

В локальных сетях, реализующих метод МДКН, как и в сетях типа ALOHA, факт приема посланных данных устанавливается с помощью подтверждения, посылаемого в виде специального пакета с приемного узла узлу-отправителю.

В случае, когда узел, пересылающий пакет, не может узнать о имеющем место в процессе передачи столкновении и продолжает передавать пакет, информационный канал сети работает вхолостую. Если же такой узел своевременно оповещен о столкновении, то коэффициент использования тракта передачи данных повышается путем прерывания передачи пакетов из всех тех узлов, которые имеют отношение к столкновению. Кроме того, если известно, что пакеты не разрушены в результате столкновения, то можно считать, что пакет достиг адресата.

Система МДКН *I*-типа, в которой предусмотрено обнаружение столкновений, называется системой *множественного доступа с контролем носителя и обнаружением столкновений* (МДКН/ОС).

Посылка и прием пакетов в локальной сети с МДКН/ОС иллюстрирует рис. 3.4. При столкновений пакетов необходима повторная их передача. Время $T_{ож}$, по истечении которого пакет посылается вторично, обычно определяется по следующим методам: с использованием константы, линейного замедления, двоичного экспоненциального алгоритма замедления.

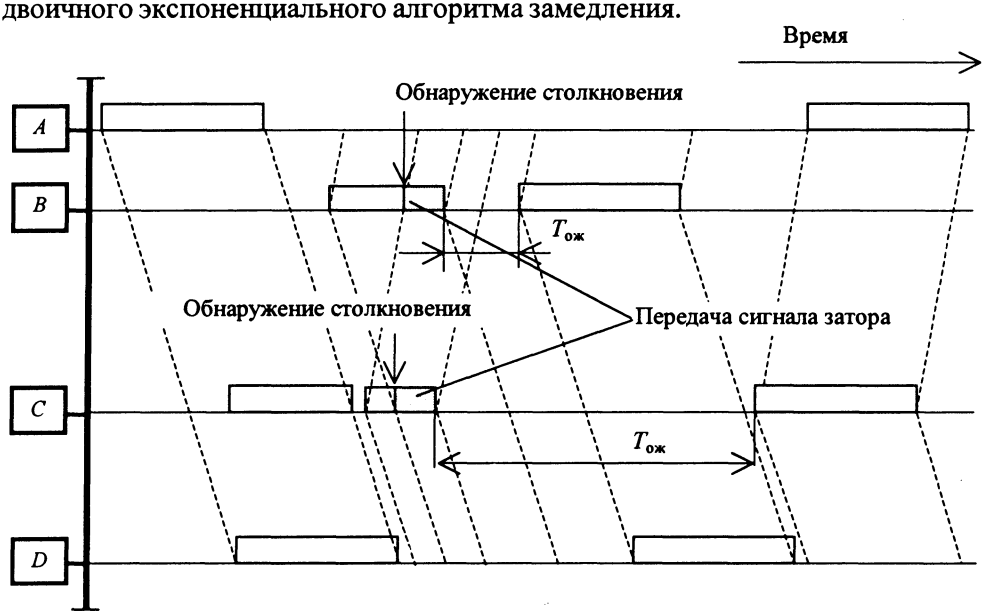


Рис. 3.4. Временна́я диаграмма МДКН/ОС

В *методе с использованием константы* (используемом в системах ALOHA и с МДКН) искомый интервал является целым кратным длительности установленного для системы *кванта* и определяется как произведение случайного числа R ($0 < R < 1$) и константы K:

$$T_{ож} = R\,K. \qquad (3.1)$$

Согласно *методу линейного замедления*, искомый интервал в квантах определяется как случайное число R, умноженное на произведение константы K и числа столкновений в конкретной передаче на рассматриваемый момент времени n:

$$T_{ож} = R\,(K\,n), \qquad (3.2)$$

здесь частоту столкновений n можно рассматривать как один из критериев, характеризующих количество запросов на передачу.

Таким образом, введение замедления при возрастании числа столкновений является своеобразной формой управления перегрузками в локальной сети.

По методу *двоичного экспоненциального алгоритма замедления* интервал ожидания повторной посылки равен:

$$T_{ож} = 2^n R\,K. \qquad (3.3)$$

Для этого метода характерно, что даже при возрастании числа запросов на передачу производительность системы не снижается, поэтому он нашел применение практически во всех локальных сетях, реализующих МДКН/ОС.

МДКН/ОС на практике оказался очень эффективным, при нем коэффициент использования тракта передачи достигает более 90 %. В локальных сетях, реализующих этот метод, не требуется специальных подтверждений приема отдельных пакетов для информирования отправителя о том, что посланный пакет не был искажен при передаче. Однако на практике имеют место случаи разрушения пакетов по различным причинам, например из-за помех. Кроме того, может оказаться, что емкость буфера недостаточна для приема пакетов, в результате чего даже при отсутствии столкновений посланные пакеты не могут использоваться абонентом, несмотря на то, что они до него дошли. Другими словами, процедуры распределения каналов и приема пакетов не всегда удачно согласованы.

Для устранения этих недостатков была разработана *система с подтверждением*, обеспечивающая следующие возможности:

- с приемом каждого информационного пакета осуществляется посылка в обратном направлении пакета с подтверждающим ответом;
- пересылка обоих пакетов (информационного и с подтверждающим ответом) по одному и тому же каналу;
- отсутствие столкновения пакета с подтверждающим ответом с другими пакетами.

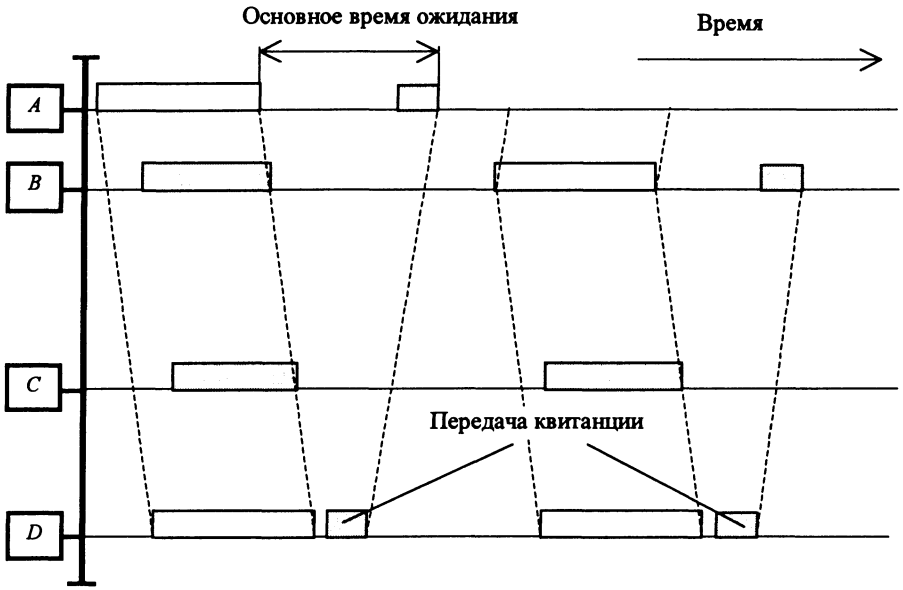


Рис. 3.5. Временна́я диаграмма МДКС/ОС с подтверждением

Узел такой системы, имеющий запросы на передачу, посылает пакеты только при возможности неоднократного использования канала после получения подтверждения о том, что он свободен, и по истечении определенной паузы, называемой *основным временем ожидания* (рис. 3.5). Это время определяется как сумма времени распространения сигнала в передающей среде в прямом и обратном направлениях и времени задержки от момента окончания приема информационного пакета до начала передачи пакета с подтверждающим ответом. Узел, принявший информационный пакет, старается как можно быстрее отослать подтверждение. Пакет с подтверждающим ответом всегда имеет приоритет перед информационным пакетом. В течение основного времени ожидания передающий узел либо принимает пакеты с подтверждением, либо ждет окончания основного времени ожидания прежде, чем принять решение о необходимости повторной посылки искаженного пакета.

Принцип работы систем с подтверждением основывается, как правило, на методах МДКН и МДКН/ОС. Примерами таких систем являются OMNINET и Acknowledging Ethernet.

3.2. Технологии локальных сетей

Технология простого Ethernet

В широком смысле под словом Ethernet понимают любой из вариантов этой технологии. В более узком смысле Ethernet – это сетевой стандарт, основанный на экспериментальной сети Ethernet Network, разработанной и реализованной фирмой Xerox в 1975 г. В 1980 г. фирмы DEC, Intel и Xerox совместно

разработали и опубликовали стандарт Ethernet версии II для сети, построенной на основе коаксиального кабеля, который стал последней версией фирменного стандарта Ethernet. Поэтому фирменную версию стандарта Ethernet называют стандартом Ethernet DIX или Ethernet II.

На основе стандарта Ethernet DIX был разработан стандарт IEEE 802.3, который во многом совпадает со своим предшественником. Различие этих стандартов состоят в следующем: в стандарте IEEE 802.3 разные уровни MAC и LLC, в оригинальном Ethernet оба эти уровня объединены в единый канальный уровень. В Ethernet DIX определяется протокол тестирования конфигурации (Ethernet Configuration Test Protocol), который отсутствует в IEEE 802.3. Несколько отличается и формат кадра, хотя минимальные и максимальные размеры кадров в этих стандартах совпадают. Часто для того, чтобы отличить Ethernet, определенный стандартом IEEE, и фирменный Ethernet DIX, первый называют технологией 802.3, а за фирменным оставляют название Ethernet без дополнительных обозначений.

В зависимости от типа физической среды стандарт IEEE 802.3 имеет модификации: 10Base-5, 10Base-2, 10Base-T, 10Base-FL, 10Base-FB. Для передачи двоичной информации по кабелю для всех вариантов физического уровня технологии Ethernet, обеспечивающих пропускную способность 10 Мбит/с, используется манчестерский код.

Все виды стандартов Ethernet основаны на одинаковом методе разделения среды передачи данных – метод доступа CSMA/CD (Carrier Sense Multiple Access / Collision Detection) и обеспечивают скорость передачи по шине 10 Мбит/с. По-русски этот метод доступа называется МДКН/ОС (множественный доступ с контролем носителя и обнаружением столкновений).

Физические спецификации технологии Ethernet по стандарту IEEE 802.3 на сегодняшний день включают следующие среды передачи данных:

- 10Base-5 – коаксиальный кабель диаметром 0,5 " («толстый» коаксиал). С волновым сопротивлением 50 Ом и максимальной длинной сегмента 500 м (без повторителей);
- 10Base-2 – коаксиальный кабель диаметром 0,25 " («тонкий» коаксиал). С волновым сопротивлением 50 Ом и максимальной длинной сегмента 185 м (без повторителей);
- 10Base-T – кабель с неэкранированной витой парой (UTP – Unshielded Twisted Pair), образующий звездообразную топологию на основе концентратора, расстояние между концентратором и конечным узлом не более 100 м.
- 10Base-F – волоконно-оптический кабель с топологией аналогичной топологии стандарта 10Base-T. Существует несколько вариантов этой спецификации: FOIRL, 10Base-FL и 10Base-FB .

Стандарт FOIRL (Fiber Optic Inter-Repeater Link) – первый стандарт комитета 802.3 для использования оптоволокна в сетях Ethernet. Он гарантирует длину оптоволоконной связи между повторителями до 1 км при общей длине сети не более 2500 м. Максимальное число повторителей между любыми уз-

лами сети равно 4. Наибольшего диаметра (2500 м) здесь достичь можно, хотя максимальные отрезки кабеля между всеми четырьмя повторителями, а также между повторителями и конечными узлами недопустимы – иначе получится сеть длиной 5000 м.

Стандарт 10Base-FL – незначительное улучшение стандарта FOIRL. Здесь повышена мощность передатчиков, поэтому максимальное расстояние между узлом и концентратором увеличилось до 2000 м. Максимальное число повторителей между узлами осталось равным 4, а максимальная длина сети – 2500 м.

Стандарт 10Base-FB предназначен только для соединения повторителей. В конечных узлах нельзя использовать этот стандарт для присоединения к портам концентратора. Между узлами сети можно установить до 5 повторителей 10Base-FB при максимальной длине одного сегмента 2000 м и максимальной длине сети 2740 м.

Повторители, соединенные по стандарту 10Base-FB, при отсутствии кадров для передачи постоянно обмениваются специальными последовательностями сигналов, отличающимися от сигналов кадров данных, для поддержания синхронизации, поэтому такие повторители вносят меньшие задержки при передаче данных из одного сегмента в другой, что позволило увеличить их число до пяти. В качестве специальных сигналов здесь используются манчестерские коды *J* и *K* в следующей последовательности: *J–J–K–K–J–J–*... Такая последовательность порождает импульсы частоты 2,5 МГц, которые и поддерживают синхронизацию приемника одного концентратора с передатчиком другого. Поэтому стандарт 10Base-FB называют также *синхронный* Ethernet.

Число 10 в указанных выше названиях обозначает битовую скорость передачи данных этих стандартов – 10 Мбит/с, а слово *Base* – метод передачи на одной базовой частоте 10 МГц (в отличие от методов, использующих несколько несущих частот, которые называются *Broadband* – широкополосными). Последний символ в названии стандарта физического уровня обозначает тип кабеля.

Протокол CSMA/CD, используемый в сетях Ethernet для разрешения конфликтов при получении доступа к среде передачи, налагает ряд ограничений на устройства и кабельную систему сетей.

- В сегменте (домен коллизий) не может находиться более 1024 устройств (DTE).

Домен коллизий (collision domain) – это часть сети Ethernet, все узлы которой распознают коллизию независимо от того, в какой части этой сети коллизия возникла. Сеть Ethernet, построенная на повторителях, всегда образует один домен коллизий. Домен коллизий соответствует одной разделяемой среде. Мосты, коммутаторы и маршрутизаторы делят сеть Ethernet на несколько доменов коллизий.

В сетях на основе коаксиальных кабелей вводятся дополнительные ограничения на число станций и протяженность кабелей.

• Время обнаружения коллизии не должно превышать времени на передачу 575 бит.

В сетях Ethernet используется множественный метод доступа к среде, позволяющий вести передачу в каждый момент только одной станции. При попытке двух или более станций начать передачу одновременно возникает конфликт доступа к среде – столкновение (коллизия). В этом случае все конфликтующие станции должны прервать передачу данных и возобновлять попытки по истечении случайного интервала времени.

Хотя в сетях Ethernet коллизии являются нормальным явлением, они увеличивают задержку и приводят к излишнему расходу полосы пропускания среды. Пакеты или их фрагменты, переданные во время конфликта, должны быть отброшены.

С ростом уровня загрузки сети (расход полосы), вероятность конфликтов возрастает. В большой сети на обнаружение коллизии, оповещение об этом сигналом «затора» и разрешение конфликта затрачивается достаточно много времени. Кроме того, на разрешение конфликтов расходуется часть полосы пропускания сетевой среды.

В соответствии со спецификациями Ethernet станция должна узнавать о возникновении конфликта до завершения передачи пакета. Поскольку длина минимального пакета с преамбулой составляет 576 бит, на обнаружение конфликта в любом случае должно затрачиваться меньшее время.

• Уменьшение интервала между пакетами на всем пути передачи не должно превышать времени на передачу 49 бит.

Промежуток времени между окончанием одного пакета и началом следующего, равный 9,6 мкс (IPG – inter packet gap), позволяет ясно различать отдельные пакеты. При передаче пакетов через повторители этот промежуток может уменьшаться. Повторитель восстанавливает синхронизацию сигналов (retiming) для устранения искажений при передаче через сетевую среду. В общем случае при восстановлении длина пакетов увеличивается за счет включения в него дополнительных битов синхронизации. Увеличение длины пакета происходит за счет сокращения IPG.

При прохождении пакета через несколько повторителей IPG может сильно уменьшиться. При слишком малом зазоре между пакетами принявшее эти пакеты устройство DTE может не успеть обработать полученный пакет к моменту прихода следующего. Исходя из этого, ограничивается протяженность самого плохого пути в сегменте так, чтобы изменение длины пакета на этом пути не превышало 49 бит. Для преодоления перечисленных ограничений используется сегментация – деление сети на меньшие фрагменты, связанные с помощью мостов, маршрутизаторов или коммутаторов.

Характеристики стандартов Ethernet приведены в табл. 3.1.

Общие ограничения для всех стандартов простого Ethernet следущие:

Номинальная пропускная способность, Мбит/с.....10

Максимальное число станций в сети 1024

Максимальное расстояние между узлами в сети, м.....2500 (в 10Base-FB 2750)

Максимальное число коаксиальных сегментов в сети.....5

Таблица 3.1. **Параметры спецификаций физического уровня для стандарта Ethernet**

Среда передачи данных	Кабель	Максимальная длина сегмента, м	Максимальное расстояние между узлами сети (при использовании повторителей), м	Максимальное число станций в сегменте	Максимальное число повторителей между любыми станциями сети
10Base-5	Толстый коаксиальный кабель RG -8 или RG11; AUI-кабель	500	2500	100	4
10Base-2	Тонкий коаксиальный кабель RG -58A/U или RG -58C/U	185	925	30	4
10Base-T	Неэкранированная витая пара категорий 3, 4, 5 (рекомендуется)	100	500	1024	4
10Base-F	Многомодовый волоконно-оптический кабель	2000	2500 (2740 для 10Base-FB)	1024	4 (5 для 10Base-FB)

Для проверки соответствия сети требованиям стандарта IEEE 802.3 необходимо начертить схему локальной сети, включив в нее все устройства с указанием длины и типа кабеля для каждого соединения, и убедиться в выполнении всех перечисленных ниже требований:

- в сети нет пути между двумя устройствами, содержащего более 5 повторителей;
- в сети не более 1024 станций (повторители не считаются);
- сеть содержит только компоненты, соответствующие стандарту IEEE 802.3, а хост-модули, концентраторы и трансиверы используют только кабели AUI, 10Base-T, FOIRL, 10Base-F, 10Base-5 или 10Base-2;

• оптические соединения имеют достаточно малое затухание, а число разъемов соответствует требованиям IEEE 802.3j;

• в сети отсутствуют соединения, превышающие предельно допустимую длину;

• пути, содержащие 3, 4 или 5 повторителей, должны удовлетворять перечисленным ниже дополнительным требованиям.

Ограничения для путей с 3 повторителями. Если самый длинный путь содержит 3 повторителя, должны выполняться следующие требования:

• между повторителями не должно быть оптических соединений длиной более 1000 м;

• между повторителями и DTE не должно быть оптических соединений длиннее 400 м;

• не должно быть соединений 10Base-T длиной свыше 100 м.

Ограничения для путей с 4 повторителями. При четырех повторителях в самом длинном пути должны выполняться следующие требования:

• между повторителями не должно быть оптических соединений длиной более 500 м;

• не должно быть соединений стандарта 10Base-T длиной свыше 100 м;

• в сети не должно быть более 3 коаксиальных сегментов с максимальной длиной кабеля.

Ограничения для путей с 5 повторителями. Если в самом длинном пути находится 5 повторителей, вводятся следующие ограничения:

• должны использоваться только оптические (FOIRL, 10Base-F) соединения или 10Base-T;

• не должно быть медных или оптических соединений с конечными станциями длиной более 100 м;

• общая длина оптических соединений между повторителями не должна превышать 2500 м (2740 м для 10Base-FB);

Приведенные способы оценки просты, но недостаточно точны. Некоторые конфигурации, не соответствующие перечисленным требованиям, оказываются совместимыми с требованиями IEEE 802.3.

Для обеспечения соответствия требованиям IEEE 802.3 в сети должны одновременно выполняться два условия:

• задержка детектирования коллизий: продолжительность пути между любыми двумя точками не должна превышать 575 бит;

• межпакетный интервал: изменение длины пакета не должно превышать 49 бит.

Стандарты 10Base-5 и 10Base-2 разрешают использование в сети не более 4 повторителей и, соответственно, не более 5 сегментов кабеля. При максимальной длине сегмента кабеля 500 м максимальная длина сети 10Base-5 составляет 2500 м. В случае стандарта 10Base-2 максимальная длина сети равна $5 \times 185 = 925$ м. Только 3 сегмента из 5 могут быть нагруженными, т. е. такими, к которым подключаются конечные станции. Правило применения повторителей в сети Ethernet называется *«правило 5–4–3»*: 5 сегментов, 4 повторителя, 3 нагруженных сегмента.

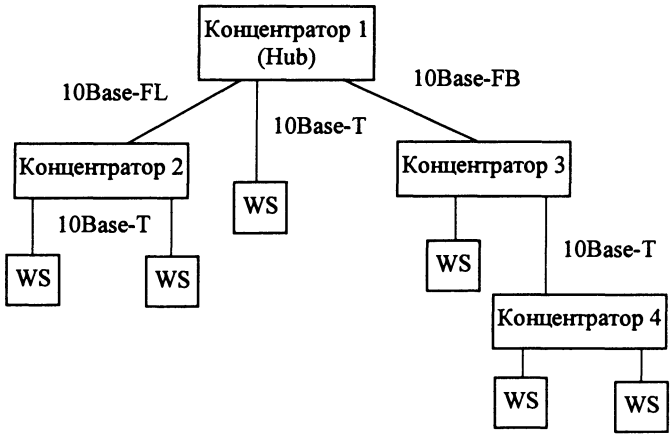


Рис. 3.6. Иерархическое соединение концентраторов Ethernet

Для обеспечения синхронизации станций при реализации процедур доступа CSMA/CD и надежного распознавания станциями коллизий в стандарте 10Base-T определено максимально число концентраторов между любыми двумя станциями сети – 4. Это правило носит название «правило 4 хабов» и оно заменяет «правило 5–4–3», применяемое к сетям 10Base-5 и 10Base-2. Очевидно, что если между любыми двумя узлами сети не должно быть больше 4 повторителей, то максимальный диаметр сети 10Base-T составляет 5 × 100 = 500 м. На рис. 3.6 представлена такая структура, образующая общую область столкновений – один домен коллизий.

При создании сети 10Base-T с большим числом станций концентраторы можно соединять друг с другом иерархическим способом, образуя древовидную структуру.

В табл. 3.2. представлены четыре основные типы кадров Ethernet.

Рассмотрим специфичные поля каждого типа кадра.

Ethernet II, разработанный первым для сетей Ethernet, дополнительно содержит поле Type – оно определяет тип протокола сетевого уровня, пакет которого переносится этим кадром (8137h – для протокола IPX, 0800h – для протокола IP, 809Bh – для протокола AppleTalk и т. д.). Все идентификаторы имеют значения старше 05DCh.

Ethernet 802.3. Этот тип кадра создан фирмой Novell, является базовым для сетей с ОС NetWare. Дополнительно содержит поле Length – длина передаваемого пакета. Поскольку в таком кадре отсутствует поле с типом протокола, то он может быть использован только для переноса IPX. Заголовок пакета IPX следует непосредственно за полем длины, поэтому первое поле пакета (поле Checksum) содержит значение FFFFh.

Ethernet 802.2, разработанный подкомитетом IEEE 802.3 в результате стандартизации сетей Ethernet, кадр содержит дополнительные поля:

Length – длина передаваемого пакета;

Таблица 3.2. **Форматы кадров простого Ethernet**

Ethernet_II	Ethernet 802.3	Ethernet 802.2	Ethernet SNAP
P (7)*	*P* (7)	*P* (7), *SFD* (1)	*P* (7), *SFD* (1)
SFD (1)	*SFD* (1)	*DA* (6)	*DA* (6), *SA* (6)
DA (6)	*DA* (6)	*SA* (6)	Length (2)
SA (6)	*SA* (6)	*Length* (2)	*DSAP* (1)
Type (2)	*Length* (2)	*DSAP* (1)	*SSAP* (1)
Пакет	*DSAP* (1)	*SSAP* (1)	*Control* (1)
(46–1500)	Пакет	*Control* (1)	*OUI* (3), *ID*(2)
	(46 – 1500)	Пакет	Пакет
FCS 4	*FCS 4*	*FCS 4*	*FCS 4*

* Цифры в круглых скобках обозначают длины полей кадров в байтах; P – преамбула – представляет собой семибайтовую последовательность единиц и нулей (101010....). Это поле предназначено для синхронизации приемной и передающей станций; SFD (*Start Frame Delimiter*) – признак начала кадра (10101011); DA (*Destination Address*), SA (*Source Address*) – адреса получателя и отправителя. Они представляют собой физические адреса сетевых адаптеров Ethernet и являются уникальными. Первые три байта адреса назначаются каждому производителю Ethernet-адаптеров (для адаптеров фирмы Intel это будет значение 00AA00h, а для адаптеров 3Com – 0020AFh), последние 3 байт определяются самим производителем. Для широковещательных кадров поле DA устанавливается в FFFFFFFFh; FCS (*Frame Check Sequence*) – контрольная сумма всех полей кадра (за исключением полей преамбулы, признака начала кадра и самой контрольной суммы). Если длина пакета передаваемых данных меньше минимальной величины, то адаптер Ethernet автоматически дополняет его до 46 байт. Этот процесс называется выравниванием (*padding*). Жесткие ограничения на минимальную длину пакета введены для обеспечения нормальной работы механизма обнаружения столкновений.

DSAP (Destination Service Access Point) – тип протокола сетевого уровня станции-получателя (E0h – для IPX),

SSAP (Source Service Access Point) – тип протокола сетевого уровня станции-отправителя,

Control – номер сегмента; используется при разбиении длинных IP-пакетов на более мелкие сегменты; для пакетов IPX это поле всегда содержит значение *03h* (обмен ненумерованными дейтаграммами).

Ethernet SNAP, являющийся модернизацией кадра Ethernet 802.2, содержит еще два поля: OUI (Organizational Unit Identifier) и ID, которые определяют тип протокола верхнего уровня SNAP Protocol ID.

Каждая станция начинает принимать кадр с преамбулы P. Затем сравнивает значение адреса DA со своим адресом. Если адреса одинаковы или пришел широковещательный кадр, или задана специальная программа обработки, то кадр копируется в буфер станции. Если нет, то кадр игнорируется.

Идентификация типа кадра сетевым адаптером осуществляется по следующему алгоритму:

- если за полем SA следует значение старше 05DCh, то это кадр Ethernet II,
- если за полем Length записан идентификатор FFFFh, то это кадр Ethernet 802.3,
- если за полем Length стоит идентификатор AAh, то это кадр Ethernet SNAP, иначе – это кадр Ethernet 802.2.

Технология Fast Ethernet

Технология Fast Ethernet является эволюционным развитием классической технологии Ethernet. Ее основными достоинствами являются:

- увеличение пропускной способности сегментов сети до 100 Мбит/с;
- сохранение метода случайного доступа Ethernet;
- сохранение звездообразной топологии сетей и поддержка традиционных сред передачи данных – витой пары и оптоволоконного кабеля.

Указанные свойства позволяют осуществлять постепенный переход от сетей 10Base-T – наиболее популярного на сегодняшний день варианта Ethernet – к скоростным сетям, сохраняющим значительную преемственность с хорошо знакомой технологией. Fast Ethernet не требует коренного переобучения персонала и замены оборудования во всех узлах сети.

Спецификация Fast Ethernet (802.3*u*), не является самостоятельным стандартом, а представляет собой дополнение к существующему стандарту 802.3 в виде глав с 21 по 30. Отличия Fast Ethernet от Ethernet сосредоточены на физическом уровне (рис. 3.7).

Более сложная структура физического уровня технологии Fast Ethernet вызвана тем, что в ней используется три варианта кабельных систем:

оптоволокно – 100Base-FX;

2-парная витая пара категории 5 – 100Base-TX;

4-парная витая пара категории 3 – 100Base-T4.

По сравнению с вариантами физической реализации Ethernet, здесь отличия каждого варианта от других глубже – меняется и количество проводников, и методы кодирования. А так как физические варианты Fast Ethernet создавались одновременно, а не эволюционно, как для сетей Ethernet, то имелась возможность детально определить неизменяемые от варианта к варианту подуровни физического уровня и остальные подуровни, специфические для каждого варианта.

Подуровни LLC и MAC в стандарте Fast Ethernet не претерпели изменений. Напомним кратко их функции.

Подуровень LLC (Logical Link Control) обеспечивает интерфейс протокола Ethernet с протоколами вышележащих уровней, например, с IP или IPX. Кадр LLC (Ethernet 802.2 без полей P, SFD, FCS по табл. 3.2), вкладывается в кадр MAC, что позволяет за счет полей DSAP и SSAP идентифицировать адрес сервисов назначения и источника соответственно. Например, при вложении в кадр LLC пакета IPX, значения как DSAP, так и SSAP должны быть равны E0. Поле управления кадра LLC позволяет реализовать процедуры обмена данными трех типов.

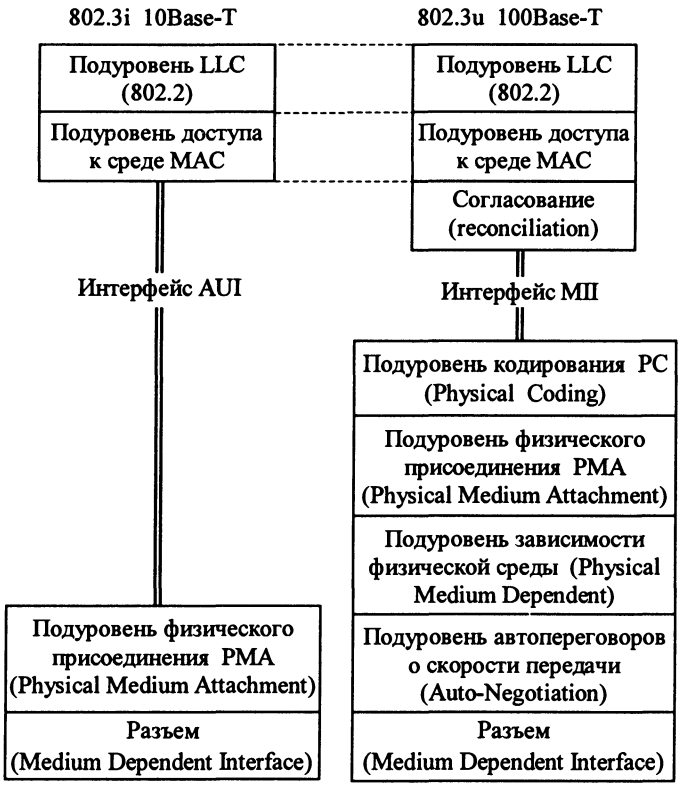


Рис. 3.7. Отличия стека протоколов 100Base-T от стека протоколов 10Base-T

Процедура 1-го типа определяет обмен данными без предварительного установления соединения и повторной передачи кадров в случае обнаружения ошибочной ситуации, т. е. является процедурой дейтаграммного типа. Именно этот тип процедуры и используется во всех практических реализациях Ethernet. Поле управления для этого типа процедур имеет значение 03, что определяет все кадры как ненумерованные.

Процедура 2-го типа определяет режим обмена с установлением соединений, нумерацией кадров, управлением потоком кадров и повторной передачей ошибочных кадров. В этом режиме протокол LLC аналогичен протоколу HDLC. В локальных сетях Ethernet такой режим используется редко.

Процедура 3-го типа определяет режим передачи данных без установления соединения, но с получением подтверждения о доставке информационного кадра адресату. Только после этого может быть отправлен следующий информационный кадр.

Существует расширение формата кадра LLC, называемое SNAP (Subnetwork Access Protocol). В случае применения расширения SNAP в поля DSAP и SSAP

записывается значение AA, тип кадра по-прежнему равен 03, а для обозначения типа протокола, вложенного в поле данных, используются следующие 4 байт, причем байты идентификатора организации (OUI) всегда равны 00 (за исключением протокола Apple Talk), а последний байт (ID) содержит идентификатор типа протокола (например, 0800 для IP).

Заголовки LLC или LLC/SNAP используют мосты и коммутаторы для трансляции протоколов канального уровня по стандарту IEEE 802.2H.

Подуровень управления доступом к среде MAC (Media Access Control) ответственен за формирование кадра Ethernet, получение доступа к разделяемой среде передачи данных и отправку с помощью физического уровня кадра по физической среде узлу назначения.

Разделяемая среда Ethernet, независимо от ее физической реализации (коаксиальный кабель, витая пара или оптоволокно с повторителями), всегда находится в одном из трех состояний: свободна, занята, коллизия. Состояние занятости соответствует нормальной передаче кадра одним из узлов сети. Состояние коллизии возникает при одновременной передаче кадров более, чем одним узлом сети.

MAC-подуровень каждого узла сети получает от физического уровня информацию о состоянии разделяемой среды. Если она свободна и у MAC-подуровня имеется кадр для передачи, то он передает его через физический уровень в сеть. Физический уровень одновременно с побитной передачей кадра следит за состоянием среды. Если за время передачи кадра коллизия не возникла, то кадр считается переданным. Если же за это время коллизия была зафиксирована, то передача кадра прекращается, и в сеть выдается специальная последовательность из 32 бит (так называемая *jam*-последовательность или сигнал «затора»), которая должна помочь однозначно распознать коллизию всеми узлами сети.

После фиксации коллизии MAC-подуровень делает случайную паузу, а затем вновь пытается передать данный кадр. Случайный характер паузы уменьшает вероятность одновременной попытки захвата разделяемой среды несколькими узлами при следующей попытке. Интервал, из которого выбирается случайная величина паузы, возрастает с каждой попыткой (до 10-й), так что при большой загрузке сети и частом возникновении коллизий происходит притормаживание узлов. Максимальное число попыток передачи одного кадра – 16, после чего MAC-подуровень оставляет данный кадр и начинает передачу следующего кадра, поступившего с LLC-подуровня.

MAC-подуровень узла приемника, получающего биты кадра от своего физического уровня, проверяет поле адреса кадра, и если адрес совпадает с его собственным, то он копирует кадр в свой буфер. Затем он проверяет, не содержит ли кадр специфические ошибки: по контрольной сумме, по максимально допустимому размеру кадра, по минимально допустимому размеру кадра, по неверно найденным границам байт. Если кадр корректен, то его поле данных передается на LLC-подуровень, если нет – то отбрасывается.

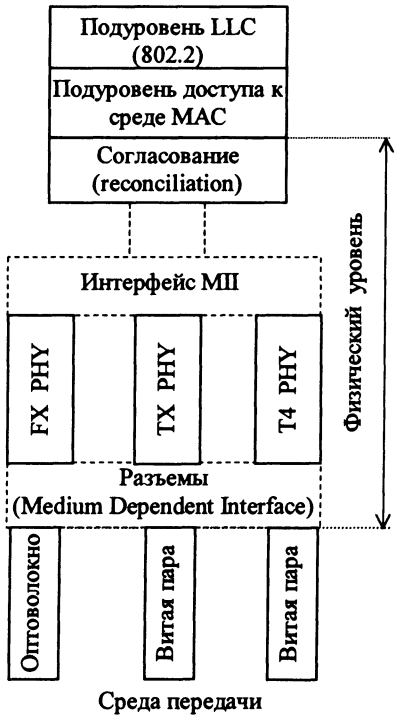


Рис. 3.8. Структура физического уровня Fast Ethernet

Форматы кадров технологии Fast Ethernet не отличаются от форматов кадров технологий простого Ethernet. Все времена передачи кадров Fast Ethernet в 10 раз меньше соответствующих времен технологии простого Ethernet:

битовый интервал составляет 10 нс вместо 100 нс;

межкадровый интервал (IPG) – 0,96 мкс вместо 9,6 мкс.

Спецификации физического уровня. Для технологии Fast Ethernet разработаны различные варианты физического уровня, отличающиеся не только типом кабеля и электрическими параметрами импульсов, как это сделано в технологии 10 Мбит/с Ethernet, но и способом кодирования сигналов и количеством используемых в кабеле проводников. Поэтому физический уровень Fast Ethernet имеет более сложную структуру, чем классический Ethernet (рис. 3.8).

Физический уровень состоит из трех подуровней:

уровень согласования (reconciliation sublayer);

независимый от среды интерфейс (MII – Media Independent Interface);

устройство физического уровня (PHY – Physical Layer Device).

Устройство физического уровня PHY обеспечивает кодирование данных, поступающих от MAC-подуровня для передачи их по кабелю определенного типа, синхронизацию передаваемых по кабелю данных, а также прием и деко-

дирование данных в узле-приемнике. Интерфейс MII поддерживает независимый от используемой физической среды способ обмена данными между MAC-подуровнем и подуровнем PHY. Этот интерфейс аналогичен по назначению интерфейсу AUI классического 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 MAC-подуровнем и подуровнями кодирования сигнала, которых в стандарте Fast Ethernet три: FX, TX и T4.

Подуровень согласования нужен для того, чтобы согласовать работу подуровня MAC с интерфейсом MII.

Интерфейс MII. Существует два варианта реализации интерфейса MII: внутренний и внешний. При внутреннем варианте микросхема, реализующая подуровни MAC и согласования, с помощью интерфейса MII соединяется с микросхемой трансивера внутри одного и того же конструктива, например, платы сетевого адаптера или модуля маршрутизатора. Микросхема трансивера реализует все функции устройства PHY.

Внешний вариант соответствует случаю, когда трансивер вынесен в отдельное устройство и соединен кабелем MII через разъем MII с микросхемой MAC-подуровня (см. рис. 1.6). Разъем MII в отличие от разъема AUI имеет 40 контактов, максимальная длина кабеля MII составляет 1 м. Сигналы, передаваемые по интерфейсу MII, имеют амплитуду 5 В.

Физический уровень 100Base-FX – многомодовое оптоволокно. Физический уровень PHY ответственен за прием данных в параллельной форме от MAC-подуровня, трансляцию их в один (TX или FX) или три последовательных потока бит с возможностью побитной синхронизации и передачу их через разъем на кабель.

Аналогично, на приемном узле уровень PHY должен принимать сигналы по кабелю, определять моменты синхронизации бит, извлекать биты из физических сигналов, преобразовывать их в параллельную форму и передавать подуровню MAC.

Между спецификациями PHY FX и PHY TX есть много общего, поэтому общие для двух спецификаций свойства будут даваться под обобщенным названием PHY FX/TX.

Структура физического уровня PHY FX включает в себя следующие подуровни:

- физического кодирования 4B/5B – PCS;
- физического присоединения PMA;
- зависимости от физической среды PMD.

Спецификация 100Base-FX определяет работу протокола Fast Ethernet по многомодовому оптоволокну в полудуплексном и полнодуплексном режимах на основе хорошо проверенной схемы кодирования и передачи оптических сигналов, использующейся уже на протяжении ряда лет в стандарте FDDI. Как и

в стандарте FDDI, каждый узел соединяется с сетью двумя оптическими волокнами, идущими от приемника (*Rx*) и от передатчика (*Tx*).

В технологии классического Ethernet для представления данных при передаче по кабелю используется манчестерское кодирование. В спецификацию PHY FX/TX без изменений перенесен метод кодирования 4B/5B, определенный в стандарте FDDI. При этом методе каждые 4 бит данных MAC-подуровня (называемых символами) представляются 5 битами потенциального кода. Потенциальные коды обладают по сравнению с манчестерскими кодами более узкой полосой спектра сигнала, а, следовательно, предъявляют меньшие требования к полосе пропускания кабеля. Кроме того, кодом 4B/5B обеспечивается синхронизация приемника с передатчиком. Коды 4B/5B построены так, что гарантируют не более трех нулей подряд при любом сочетании бит в исходной информации.

Так как исходные биты MAC-подуровня должны передаваться со скоростью 100 Мбит/с, то наличие одного избыточного бита вынуждает передавать биты результирующего кода 4B/5B со скоростью 125 Мбит/с, т. е. межбитовое расстояние в устройстве PHY составляет 8 нс.

Поскольку из 32 возможных комбинаций кода 4B/5B для кодирования исходных данных нужно только 16, то остальные 16 комбинаций используются в служебных целях. Наличие служебных символов позволило применить в спецификациях FX/TX схему непрерывного обмена сигналами между передатчиком и приемником и при свободном состоянии среды, что отличает их от спецификации 10Base-T, когда незанятое состояние среды обозначается полным отсутствием на ней импульсов информации. Для обозначения незанятого состояния среды используется служебный символ Idle (11111), который постоянно циркулирует между передатчиком и приемником, поддерживая их синхронизм и в периодах между передачами информации, а также позволяя контролировать физическое состояние линии.

Существование запрещенных комбинаций символов позволяет отбраковывать ошибочные символы, что повышает устойчивость работы сетей с PHY FX/TX. Для отделения кадра Ethernet от символов Idle используется комбинация символов Start Delimiter (пара символов JK), а после завершения кадра перед первым символом Idle вставляется символ T.

После преобразования 4-битовых порций MAC-кодов в 5-битовые порции PHY их необходимо представить в виде оптических или электрических сигналов в кабеле, соединяющем узлы сети. Спецификации PHY FX и PHY TX используют для этого различные методы физического кодирования (NRZI и MLT-3 соответственно). Эти же методы определены в стандарте FDDI для передачи сигналов по оптоволокну (спецификация PMD) и витой паре (спецификация TP-PMD).

Физический уровень 100Base-TX – двухпарная витая пара. Основные отличия этого уровня от спецификации PHY FX состоят в использовании метода MLT-3 для передачи сигналов 5-битовых порций кода 4B/5B по витой паре и

наличии функции автопереговоров (Auto-negotiation) для выбора режима работы порта. Метод MLT-3 использует потенциальные сигналы двух полярностей для представления 5-битовых порций информации.

Кроме применения метода MLT-3, спецификация PHY TX отличается от спецификации PHY FX тем, что в ней предусмотрена пара скрэмблер/дескрэмблер (scrambler/descrambler), как это определено в спецификации ANSI TP-PMD. Скрэмблер принимает 5-битовые порции данных от подуровня PCS, выполняющего кодирование 4B/5B, и кодирует сигналы перед передачей на подуровень MLT-3 таким образом, чтобы равномерно распределить энергию сигнала по всему частотному спектру. Это уменьшает электромагнитное излучение кабеля.

Спецификации PHY TX и PHY T4 поддерживают функцию Auto-negotiation, с помощью которой два взаимодействующих устройства PHY могут автоматически выбрать наиболее эффективный режим работы.

В настоящее время определено 5 различных режимов работы, которые могут поддерживать устройства PHY TX или PHY T4 на витых парах:

10Base-T (2 пары категории 3);

10Base-T full-duplex (2 пары категории 3);

100Base-TX (2 пары категории 5 (или Type 1A STP);

100Base-TX full-duplex (2 пары категории 5 (или Type 1A STP);

100Base-T4 (4 пары категории 3).

Режим 10Base-T имеет самый низкий приоритет при переговорном процессе, а режим 100Base-T4 – самый высокий. Переговорный процесс происходит при включении питания устройства или может быть инициирован в любой момент модулем управления.

Узлы, поддерживающие спецификации PHY FX и PHY TX, могут работать в полнодуплексном режиме (full-duplex mode). В этом режиме не используется метод доступа к среде CSMA/CD и отсутствует понятие коллизий – каждый узел одновременно передает и принимает кадры данных по каналам *Tx* и *Rx*. Полнодуплексная работа возможна только при соединении сетевого адаптера с коммутатором или же при непосредственном соединении коммутаторов.

При полнодуплексной работе стандарты 100Base-TX и 100Base-FX обеспечивают скорость обмена данными между узлами 200 Мбит/с.

Физический уровень 100Base-T4 – четырехпарная витая пара. Спецификация PHY T4 была разработана для возможности использования для высокоскоростного Ethernet имеющуюся проводку на витой паре категории 3. Чтобы повысить общую пропускную способность за счет одновременной передачи потоков бит по нескольким витым парам эта спецификация использует все 4 пары кабеля.

Вместо кодирования 4B/5B в этом методе используется кодирование 8B/6T. Каждые 8 бит информации MAC-уровня кодируются шестью троичными цифрами (ternary symbols), т. е. цифрами, имеющими три состояния. Каждая тро-

ичная цифра имеет длительность 40 нс. Группа из 6 троичных цифр затем передается на одну из трех передающих витых пар, независимо и последовательно. Четвертая пара используется для прослушивания несущей частоты в целях обнаружения коллизии. Скорость передачи данных по каждой из трех передающих пар равна 33,3 Мбит/с, следовательно общая скорость протокола 100Base-T4 составляет 100 Мбит/с. В то же время из-за принятого способа кодирования скорость изменения сигнала на каждой паре равна всего 25 Мбод, что и позволяет использовать витую пару категории 3.

Правила построения сегментов сети по технологии Fast Ethernet. Технология Fast Ethernet, как и все некоаксиальные варианты Ethernet, рассчитана на подключение конечных узлов (компьютеров с соответствующими сетевыми адаптерами) к многопортовым концентраторам-повторителям или коммутаторам.

Правила корректного построения сегментов сетей Fast Ethernet включают:

- ограничения на максимальные длины сегментов, соединяющих DTE с DTE;
- ограничения на максимальные длины сегментов, соединяющих DTE с портом повторителя;
- ограничения на максимальный диаметр сети;
- ограничения на максимальное число повторителей и максимальную длину сегмента, соединяющего повторители.

Ограничения длин сегментов DTE-DTE. В качестве DTE (Data Terminal Equipment) может выступать любой источник кадров данных для сети: сетевой адаптер, порт моста, порт маршрутизатора, модуль управления сетью и другие подобные устройства. Порт повторителя не является DTE. В типичной конфигурации сети Fast Ethernet несколько DTE подключается к портам повторителя, образуя сеть звездообразной топологии.

Спецификация IEEE 802.3*u* определяет максимальную длину сегментов DTE-DTE:

100Base-TX	(кабель *Category* 5 *UTP*)	100 м
100Base-FX	(многомодовое волокно 62,5/125 мкм)	412 м (полудуплекс) 2 км (полный дуплекс)
100Base-T4	(кабель Category 3, 4 или 5 *UTP*)	100 м

Ограничения, связанные с соединениями с повторителями. Повторители Fast Ethernet делятся на два класса.

Повторители класса I поддерживают все типы систем кодирования физического уровня: 100Base-TX/FX и 100Base-T4.

Повторители класса II поддерживают только один тип системы кодирования физического уровня – 100Base-TX/FX или 100Base-T4.

В одном домене коллизий допускается наличие только одного повторителя класса I. Это связано с тем, что такой повторитель вносит большую задержку при распространении сигналов из-за необходимости трансляции различных систем сигнализации.

Повторители класса II в домене коллизий соединяются между собой кабелем не длиннее 5 м, а их число не превышает 2.

Небольшое количество повторителей Fast Ethernet не является серьезным препятствием при построении сетей. Во-первых, наличие стековых повторителей снимает проблемы ограниченного числа портов: все каскадируемые повторители представляют собой один повторитель с достаточным числом портов – до нескольких сотен. Во-вторых, применение коммутаторов и маршрутизаторов делит сеть на несколько доменов коллизий с небольшим числом станций в каждом.

В табл. 3.3 сведены правила построения сети на основе повторителей класса I.

Таблица 3.3. **Правила построения сети на основе повторителей класса I**

Тип кабеля	Максимальный диаметр сети, м	Максимальная длина сегмента, м
Только витая пара (*TX*)	200	100
Только оптоволокно (*FX*)	272	136
Несколько сегментов на витой паре и один сегмент на оптоволокне	260	100 (*TX*) 160 (*FX*)
Несколько сегментов на витой паре и несколько сегментов на оптоволокне	272	100 (*TX*) 136 (*FX*)

Fast Ethernet следует применять в организациях и частях сетей, где до этого широко применялся простой Ethernet, но сегодняшние условия или же ближайшие перспективы требуют в этих частях сетей более высокой пропускной способности. Однако технология Fast Ethernet кроме положительных свойств, унаследовала и недостатки технологии Ethernet:

- большие задержки доступа к среде при коэффициенте использования среды выше 30...40 %, являющиеся следствием применения алгоритма доступа CSMA/CD;
- небольшие расстояния между узлами даже при использования оптоволокна – следствие метода обнаружения коллизий;
- отсутствие определения избыточных связей в стандарте и отсутствие поддержки приоритетного трафика приложений реального времени.

Технология *Gigabit Ethernet*

Основная идея разработчиков стандарта 802.3z Gigabit Ethernet состоит в максимальном сохранении идей классической технологии Ethernet при достижении битовой скорости в 1000 Мбит/с.

Gigabit Ethernet, так же как и его менее скоростные собратья, на уровне протокола не поддерживает:

качество обслуживания;

избыточные связи;

тестирование работоспособности узлов и оборудования (в последнем случае – за исключением тестирования связи порт-порт, как это делается в Ethernet 10Base-T и 10Base-F и Fast Ethernet).

В технологии Gigabit Ethernet по сравнению с технологиями Ethernet и Fast Ethernet:

- сохраняются все форматы кадров Ethernet;
- сохраняется полудуплексная версия протокола, поддерживающая метод доступа CSMA/CD, и полнодуплексная версия, работающая с коммутаторами;
- поддерживаются все основные виды кабелей, используемых в Ethernet и Fast Ethernet: волоконно-оптический, витая пара категории 5 и коаксиальный кабель.

На рис. 3.9 показана структура уровней Gigabit Ethernet. Как и в стандарте Fast Ethernet, в Gigabit Ethernet не существует универсальной схемы кодирования сигнала, которая была бы идеальной для всех физических интерфейсов – так, с одной стороны, для стандартов 1000Base-LX/SX/CX используется кодирование 8B/10B, а с другой стороны, для стандарта 1000Base-T используется специальный расширенный линейный код TX/T2. Функцию кодирования выполняет подуровень кодирования PCS, размещенный ниже средонезависимого интерфейса GMII (Gigabit Media Independent Interface).

Интерфейс GMII. Он обеспечивает взаимодействие между уровнем MAC и физическим уровнем, является расширением интерфейса МП и может поддерживать скорости 10, 100 и 1000 Мбит/с; имеет отдельные 8-разрядные

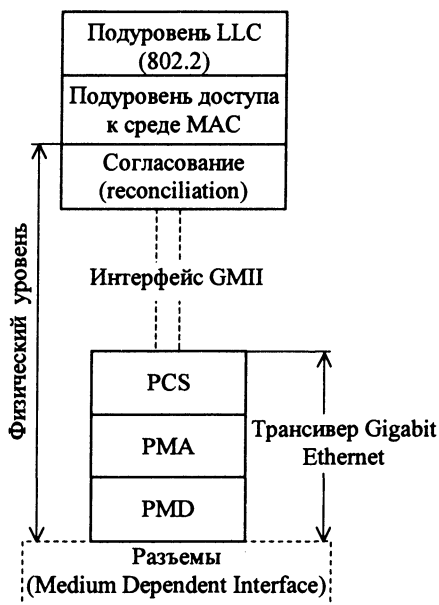


Рис. 3.9. Структура уровней стандарта Gigabit Ethernet

приемник и передатчик и может поддерживать полудуплексный и дуплексный режимы. Кроме этого, GMII интерфейс имеет одну сигнальную цепь, обеспечивающую синхронизацию, две сигнальных цепи состояния линии: первая указывает наличие несущей, а вторая – отсутствие коллизий, а также несколько других сигнальных цепей и питание. Трансиверный модуль, охватывающий физический уровень и обеспечивающий один из физических средозависимых интерфейсов, может подключаться, например, к коммутатору Gigabit Ethernet посредством GMII интерфейса.

Подуровень физического кодирования PCS. При подключении интерфейсов группы 1000Base-X подуровень PCS использует блочное избыточное кодирование 8B/10B, заимствованное из стандарта ANSI X3T11 Fibre Channel. В подуровне PCS каждые 8 входных битов, предназначенных для передачи на удаленный узел, преобразовываются в 10-битные символы. Кроме этого, в выходном последовательном потоке присутствуют специальные контрольные 10-битные символы, используемые, например, для расширения носителя (дополняют кадр Gigabit Ethernet до его минимально размера 512 байт).

При подключении интерфейса 1000Base-T подуровень PCS осуществляет специальное помехоустойчивое кодирование для обеспечения передачи по витой паре UTP Cat.5 на расстояние до 100 м – линейный код TX/T2, разработанный компанией Level One Communications.

Два сигнала состояния линии – наличие несущей и отсутствие коллизий – генерируются этим подуровнем.

Подуровни PMA и PMD. Физический уровень Gigabit Ethernet использует несколько интерфейсов, включая традиционную витую пару категории 5, а также многомодовое и одномодовое волокна.

Подуровень PMA преобразует параллельный поток символов от PCS в последовательный поток, и выполняет обратное преобразование (распараллеливание) входящего последовательного потока от PMD. Подуровень PMD определяет оптические/электрические характеристики физических сигналов для разных сред. Всего определены 4 типа физических интерфейсов среды (рис. 3.10), которые отражены в спецификациях стандарта 802.3z (1000Base-X) и 802.3ab (1000Base-T).

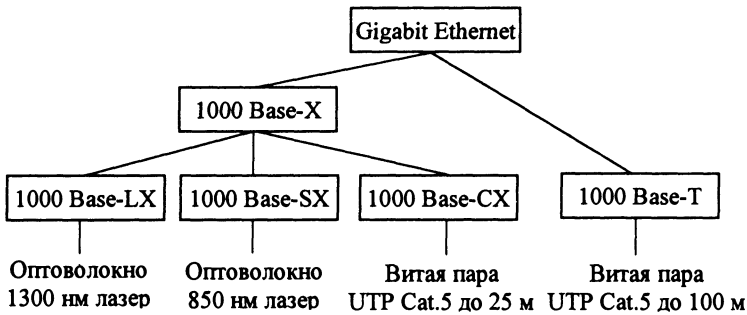


Рис. 3.10. Физические интерфейсы стандарта Gigabit Ethernet

1000Base-X основывается на стандарте физического уровня Fibre Channel – технологии взаимодействия рабочих станций, суперкомпьютеров, устройств хранения и периферийных узлов. Fibre Channel имеет 4-уровневую архитектуру. Два нижних уровня *FC-0* (интерфейсы и среда) и *FC-1* (кодирование/декодирование) перенесены в Gigabit Ethernet. Поскольку Fibre Channel является проверенной технологией, то это значительно сократило время на разработку оригинального стандарта Gigabit Ethernet.

Блочный код 8B/10B аналогичен коду 4B/5B, принятому в стандарте FDDI. Однако код 4B/5B не применяется в Fibre Channel, потому что он не обеспечивает баланса по постоянному току (хотя код 4B/5B не обеспечивает баланса по постоянному току, в стандарте FDDI предусмотрен специальный дополнительный узел, поддерживающий баланс по постоянному току с дрейфом в пределах ±10 %. При использовании кода 8B/10B необходимость в таком узле отпадает, и при этом полностью отсутствует дрейф постоянной составляющей). Отсутствие баланса потенциально может привести к нагреванию лазерных диодов, зависящему от передаваемых данных, что может быть причиной дополнительных ошибок при высоких скоростях передачи.

Спецификация 1000Base-X подразделяется на три физических интерфейса:

• 1000Base-SX – определяет лазеры с допустимой длиной излучения в диапазоне 770...860 нм, с мощностью излучения передатчика от –10 до 0 дБм, при отношении ON/OFF (сигнал / нет сигнала) не менее 9 дБ. Чувствительность приемника составляет 17 дБм, его насыщение – 0 дБм;

• 1000Base-LX – определяет лазеры с допустимой длиной излучения в диапазоне 1270...1355 нм, с мощностью излучения передатчика от –13,5 до –3 дБм, при отношении ON/OFF (есть сигнал / нет сигнала) не менее 9 дБ. Чувствительность приемника составляет 19 дБм, его насыщение –3 дБм;

• 1000Base-CX – экранированная витая пара (STP «twinax») на короткие расстояния.

Поддерживаемые расстояния для стандартов 1000Base-X приведены в табл. 3.4.

При кодировании кодом 8B/10B битовая скорость в оптической линии составляет 1250 бит/с. Это означает, что полоса пропускания участка кабеля допустимой длины должна превышать 625 МГц. Из табл. 3.4 следует, что этот критерий для строчек 2–6 выполняется. Из-за большой скорости передачи Gigabit Ethernet следует быть внимательным при построении протяженных сегментов. Безусловно, предпочтение отдается одномодовому волокну. При этом характеристики оптических приемопередатчиков могут быть значительно выше. Например, компания NBase выпускает коммутаторы с портами Gigabit Ethernet, обеспечивающими расстояния до 40 км по одномодовому волокну без ретрансляции (используются узкоспектральные DFB-лазеры, работающие на длине волны 1550 нм).

Таблица 3.4. **Поддерживаемые расстояния для стандартов1000Base-X**

Стандарт	Тип волокна/медного кабеля	Полоса пропускания (не хуже), МГц на 1 км	Максимальное расстояние*, м
1000Base-LX (лазерный диод 1300 нм)	Одномодовое волокно (9 мкм)	–	5000**
	Многомодовое волокно (50 мкм)***	500	550
	Многомодовое волокно (62,5 мкм)***	320	400
1000Base-SX (лазерный диод 850 нм)	Многомодовое волокно (50 мкм)	400	500
	Многомодовое волокно (62,5 мкм)	200	275
	Многомодовое волокно (62,5 мкм)	160	220
1000Base-CX	Экранированная витая пара: STP 150 Ом	–	25

* Все расстояния, за исключением последнего (25 м), предполагают использование дуплексного режима.

** Большее расстояние может обеспечивать оборудование некоторых производителей, оптические сегменты без промежуточных ретрансляторов/усилителей могут достигать 100 км.

*** Может потребоваться специальный переходной шнур.

Технология Gigabit Ethernet для передачи по неэкранированной витой паре категории 5 на расстояния до 100 м использует все четыре пары медного кабеля. Скорость передачи по одной паре составляет 250 Мбит/с. Влияние ближних и дальних переходных помех от трех соседних витых пар на данную пару в четырехпарном кабеле требует разработки специальной скрэмблированной помехоустойчивой передачи, интеллектуального узла распознавания и восстановления сигнала на приеме.

Одним из методов физического кодирования является 5-уровневое импульсно-амплитудное кодирование PAM-5. Идея его заключается в следующем.

Для кодирования данных код PAM-5 использует 5 уровней потенциала: –2, –1, 0, +1, +2. Поэтому за один такт по одной паре передается 2,322 бит информации. При этом если передавать 8 бит за такт (по 4 парам), то выдерживается требуемая скорость передачи в 1000 Мбит/с. Пятый уровень добавлен для создания избыточности кода. Так как код PAM-5 содержит $5^4 = 625$ комбинаций и если передавать за один такт по всем четырем парам 8 бит данных, то для этого требуется всего $2^8 = 256$ комбинаций, что дает дополнительный резерв 6 дБ в соотношении сигнал/шум. Оставшиеся комбинации приемник может использовать для контроля принимаемой информации и выделения правильных комбинаций на фоне шума. Код PAM-5 на тактовой частоте 125 МГц укладывается в полосу 100 МГц кабеля категории 5.

Подуровень MAC стандарта Gigabit Ethernet использует тот же самый протокол передачи CSMA/CD, что и Ethernet и Fast Ethernet. Основные ограничения на максимальную длину сегмента (или коллизионного домена) определяются этим протоколом.

В стандарте Ethernet IEEE 802.3 принят минимальный размер кадра 64 байт. Именно значение минимального размера кадра определяет максимальное допус-

тимое расстояние между станциями (диаметр коллизионного домена). Время, в течение которого станция передает такой кадр (время канала) равно 512 битовым интервалам (BT) или 51,2 мкс. Максимальная длина сети Ethernet определяется из условия разрешения коллизий, а именно временем, за которое сигнал доходит до удаленного узла и возвращается обратно, не должно превышать 512 BT (без учета преамбулы).

При переходе от Ethernet к Fast Ethernet скорость передачи возрастает, а время трансляции кадра длиной 64 байт соответственно сокращается, оно равно 512 BT или 5,12 мкс (в Fast Ethernet 1 BT = 0,01 мкс). Для того чтобы можно было обнаруживать все коллизии до конца передачи кадра, необходимо выполнение одного из условий:

сохранить *прежнюю максимальную длину сегмента,* но увеличить *время канала* (и следовательно, увеличить минимальную длину кадра);

сохранить *время канала* (прежний размер кадра), но уменьшить *максимальную длину сегмента.*

В Fast Ethernet сохранен такой же минимальный размер кадра, как в Ethernet. Это обеспечило совместимость, но привело к значительному уменьшению диаметра коллизионного домена.

В силу преемственности стандарт Gigabit Ethernet должен поддерживать те же минимальный и максимальный размеры кадра, которые приняты в Ethernet и Fast Ethernet. Но поскольку скорость передачи возрастает, то, соответственно, уменьшается и время передачи пакета аналогичной длины. При сохранении прежней минимальной длины кадра это привело бы к уменьшению диаметра сети до 20 м, что мало пригодно. Поэтому при разработке стандарта Gigabit Ethernet было увеличено *время канала* до 512 байтовых интервалов, что в 8 раз превосходит время канала Ethernet и Fast Ethernet. Чтобы поддержать совместимость со стандартами Ethernet и Fast Ethernet, минимальный размер кадра не увеличили, а добавили к нему дополнительное поле, называемое *расширением носителя* (carrier extension). Символы в дополнительном поле обычно не несут служебной информации, но они заполняют канал и увеличивают «коллизионное окно». В результате коллизия будет регистрироваться всеми станциями при большем диаметре коллизионного домена. Если станция желает передать короткий (менее 512 байт) кадр, то при передаче добавляется это поле (расширение носителя), дополняющее кадр до 512 байт. Поле контрольной суммы вычисляется только для оригинального кадра и не распространяется на поле расширения. При приеме кадра поле расширения отбрасывается, и уровень LLC не знает о нем. Если размер кадра равен или превосходит 512 байт, то поле расширения носителя отсутствует. Расширение носителя – это наиболее естественное решение, которое позволило сохранить совместимость со стандартом Fast Ethernet и такой же диаметр коллизионного домена, но оно привело к излишней трате полосы пропускания. До 448 байт (512 – 64) может расходоваться вхолостую при передаче короткого кадра. На стадии разработки стандарта Gigabit Ethernet компанией NBase Communications было внесено предложение по модернизации стандарта. Эта модернизация, получившая название *пакетная перегруженность* (packet bursting), позволяет эффективней исполь-

зовать поле расширения. Если у станции/коммутатора имеется несколько небольших кадров для отправки, то первый кадр дополняется полем расширения носителя до 512 байт и отправляется. Остальные кадры отправляются вслед с минимальным межкадровым интервалом в 96 бит, с одним важным исключением – межкадровый интервал заполняется символами расширения. Таким образом, между посылками коротких оригинальных кадров в среде продолжают передаваться сигналы и никакое другое устройство сети не может вклиниться в передачу. Подобное пристраивание кадров может происходить до тех пор, пока полное число переданных байт не превысит 1518. Пакетная перегруженность уменьшает вероятность образования коллизий, что, безусловно, увеличивает производительность сети, особенно при больших нагрузках.

Технология 100VG-AnyLAN

В качестве альтернативы технологии Fast Ethernet, фирмы AT&T и HP выдвинули проект новой технологии со скоростью передачи данных 100 Мбит/с – 100Base-VG. В этом проекте было предложено усовершенствовать метод доступа с учетом потребности мультимедийных приложений и при этом сохранить совместимость формата пакета с форматом пакета сетей 802.3. В сентябре 1993 г. по инициативе фирм IBM и HP был образован комитет IEEE 802.12, который занялся стандартизацией новой технологии. Проект был расширен за счет поддержки в одной сети кадров не только формата Ethernet, но и формата Token Ring. В результате новая технология получила название 100VG-AnyLAN, т. е. технология для любых сетей (Any LAN – любые сети), имея в виду, что в локальных сетях технологии Ethernet и Token Ring используются в подавляющем количестве узлов. В 1995 г. технология 100VG-AnyLAN получила статус стандарта IEEE 802.12.

В технологии 100VG-AnyLAN определены новый метод доступа Demand Priority и новая схема квартетного кодирования Quartet Coding, использующая избыточный код 5B/6B.

Метод доступа Demand Priority основан на передаче концентратору функций арбитра, решающего проблему доступа к разделяемой среде. Метод Demand Priority повышает пропускную способность сети за счет введения детерминированного доступа к общей среде, использующего два уровня приоритетов: низкий – для обычных приложений и высокий – для мультимедийных.

Структура сети 100VG-AnyLAN. Сеть 100VG-AnyLAN всегда включает центральный концентратор, называемый концентратором уровня 1 или корневым концентратором (рисунок 3.11). Корневой концентратор имеет связи с каждым узлом сети, образуя топологию «звезда». Он представляет собой интеллектуальный центральный контроллер, который управляет доступом к сети, постоянно выполняя цикл кругового сканирования своих портов и проверяя наличие запросов на передачу кадров от присоединенных к ним узлов. Концентратор принимает кадр от узла, выдавшего запрос, и передает его только через тот порт, к которому присоединен узел с адресом, совпадающим с адресом назначения, указанным в кадре.

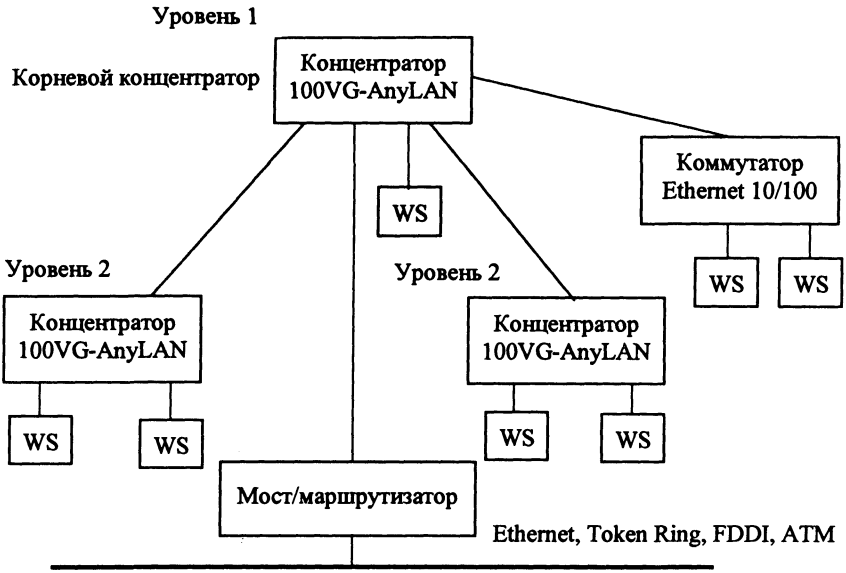


Рис. 3.11. Структура сети 100VG-AnyLAN

Каждый концентратор может быть сконфигурирован на поддержку либо кадров 802.3 Ethernet, либо кадров 802.5 Token Ring. Все концентраторы, расположенные в одном и том же логическом сегменте (не разделенном мостами, коммутаторами или маршрутизаторами), должны быть сконфигурированы на поддержку кадров одного типа. Для соединения сетей 100VG-AnyLAN (рис. 3.11), использующих разные форматы кадров 802.3, необходим мост, коммутатор или маршрутизатор. Аналогичное устройство требуется и в том случае, когда сеть 100VG-AnyLAN соединяется с сетью FDDI или ATM.

Каждый концентратор имеет один «восходящий» (up-link) порт и *N* «нисходящих» портов (down-link), как это показано на рис. 3.12.

Восходящий порт работает как порт узла, но он зарезервирован для присоединения в качестве узла к концентратору более высокого уровня. Нисходящие порты служат для присоединения узлов, в том числе и концентраторов нижнего уровня. Каждый порт концентратора может быть сконфигурирован для работы в нормальном режиме или в режиме монитора. Порт, сконфигурированный для работы в нормальном режиме, передает только те кадры, которые предназначены узлу, подключенному к данному порту. Порт, сконфигурированный для работы в режиме монитора, передает все кадры, обрабатываемые концентратором. Такой порт может использоваться для подключения анализатора протоколов.

Узел представляет собой компьютер или коммуникационное устройство технологии 100VG-AnyLAN: мост, коммутатор, маршрутизатор или концент-

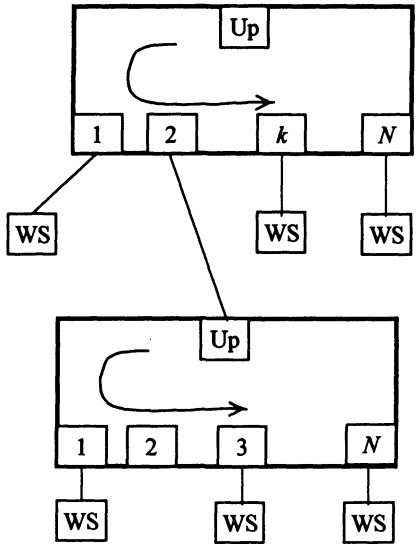


Рис. 3.12. Круговой опрос портов концентраторами сети 100VG-AnyLAN

ратор. Концентраторы, подключаемые как узлы, называются концентраторами 2- и 3-го уровней. Разрешается образовывать до трех уровней иерархии концентраторов.

Связь, соединяющая концентратор и узел, может быть образована либо 4 парами неэкранированной витой пары категорий 3, 4 или 5 (4UTP Cat 3, 4, 5), либо 2 парами неэкранированной витой пары категории 5 (2UTP Cat 5), либо 2 парами экранированной витой пары типа 1 (2STP Type 1), либо 2 парами многомодового оптоволоконного кабеля.

Варианты кабельной системы можно использовать любые. Наибольшее распространение получил первый разработанный вариант 4UTP.

В табл. 3.5, составленной по материалам компании Hewlett-Packard, приведены результаты сравнения этой технологии с технологиями 10Base-T и 100Base-T.

Структура стека протоколов технологии 100VG-AnyLAN согласуется с архитектурными моделями OSI/ISO и IEEE, в которых канальный уровень разделен на подуровни. Стек протоколов технологии 100VG-AnyLAN состоит из подуровня доступа к среде (MAC – Media Access Control), подуровня, не зависящего от физической среды (PMI – Physical Media Independent) и подуровня, зависящего от физической среды (PMD – Physical Media Dependent).

Функции уровня MAC включают реализацию протокола доступа Demand Priority, подготовку линии связи и формирования кадра соответствующего формата.

Таблица 3.5. **Сравнение технологий**

Характеристика	Технология		
	10Base-T	100VG-AnyLAN	100Base-T
Топология			
Максимальный диаметр сети, м	2500	8000	412
Каскадирование концентраторов	3 уровня	5 уровней	2 концентратора
Кабельная система			
UTP Cat 3,4, м	100	100	100
UTP Cat 5, м	150	200	100
STP Type 1, м	100	100	100
Оптоволокно, м	2000	2000	412
Производительность			
При длине сети 100 м, %	80 (теоретическая)	95 (продемонстрированная)	80 (теоретическая)
При длине сети 2500 м, %	80 (теоретическая)	80 (продемонстрированная)	Не поддерживается
Технология			
Кадры IEEE 802.3	+	+	+
Кадры 802.5	–	+	–
Метод доступа	CSMA/CD	Demand Priority	CSMA/CD + подуровень согласования

Метод Demand Priority (приоритетный доступ по требованию) основан на том, что узел, которому нужно передать кадр по сети, передает запрос (требование) на выполнение этой операции концентратору. Каждый запрос может иметь либо низкий, либо высокий приоритеты. Высокий приоритет отводиться для трафика чувствительных к задержкам мультимедийных приложений.

Высокоприоритетные запросы всегда обслуживаются раньше низкоприоритетных. Требуемый уровень приоритета кадра устанавливается протоколами верхних уровней, не входящими в технологию 100VG-AnyLAN, например, Real Audio, и передается для отработки уровню MAC.

Как показано на рис. 3.12, концентратор уровня 1 постоянно сканирует запросы узлов, используя алгоритм кругового опроса (round-robin). Это сканирование позволяет концентратору определить, какие узлы требуют передачи кадров через сеть и каковы их приоритеты.

В течение одного цикла кругового сканирования каждому узлу разрешается передать один кадр данных через сеть. Концентраторы, присоединенные как узлы к концентраторам верхних уровней иерархии, также выполняют свои циклы сканирования и передают запрос на передачу кадров концентратору. Концентратор нижнего уровня с N портами имеет право передать N кадров в течение одного цикла опроса.

Каждый концентратор ведет отдельные очереди для низкоприоритетных и высокоприоритетных запросов. Низкоприоритетные запросы обслуживаются только до тех пор, пока не получен высокоприоритетный запрос. В этом случае текущая передача низкоприоритетного кадра завершается и обрабатывается высокоприоритетный запрос. Перед возвратом к обслуживанию низкоприоритетных кадров должны быть обслужены все высокоприоритетные запросы. Чтобы гарантировать доступ для низкоприоритетных запросов в периоды высокой интенсивности поступления высокоприоритетных запросов, вводится порог ожидания запроса. Если у какого-либо низкоприоритетного запроса время ожидания превышает этот порог, то ему присваивается высокий приоритет.

Пример. На рис. 3.12 показан цикл кругового опроса. Предположим, что все порты передали запросы нормального приоритета и в начальный момент времени корневой концентратор начал круговой опрос. Порядок обслуживания портов будет следующим: 1–1 (уровень 1 – порт 1), 2–1, 2–3, 2–*N*, 1–3, 1–*N*.

Если предположить, что узлы 1–1, 2–3 и 1–3 выставили высокоприоритетные запросы. В этом случае порядок обслуживания будет таким: 1–1, 2–3, 1–3, 2–1, 2–*N*, 1–*N*.

Процедура подготовки линии Link Training «обучает» внутренние схемы концентратора и узла приему и передаче данных, а также проверяет работоспособность линии, соединяющей концентратор и узел. Во время подготовки линии концентратор и узел обмениваются серией специальных тестовых кадров. Данная процедура включает функциональный тест кабеля, дающий возможность убедиться в том, что кабель правильно соединяет контакты разъемов и что информация может быть корректно передана между концентратором и узлом. Процедура подготовки также позволяет концентратору автоматически узнать информацию об узлах, подключенных к каждому порту. Кадры, получаемые концентратором от узла во время подготовки, содержат данные о типе устройства (конечный узел, концентратор, мост, маршрутизатор, анализатор протокола и т. п.), режиме работы (нормальный или монитор), адрес узла, присоединенного к данному порту.

Процедура подготовки инициируется узлом, когда узел или концентратор впервые включаются, или при первом присоединении узла к концентратору. Узел или концентратор могут потребовать выполнения процедуры подготовки при обнаружении ошибочной ситуации.

Уровень MAC получает кадр от уровня LLC и добавляет к нему адрес узла-источника, дополняет поле данных байтами-заполнителями до минимально допустимого размера, если это требуется, а затем вычисляет контрольную сумму и помещает ее в соответствующее поле. После этого кадр передается на физический уровень.

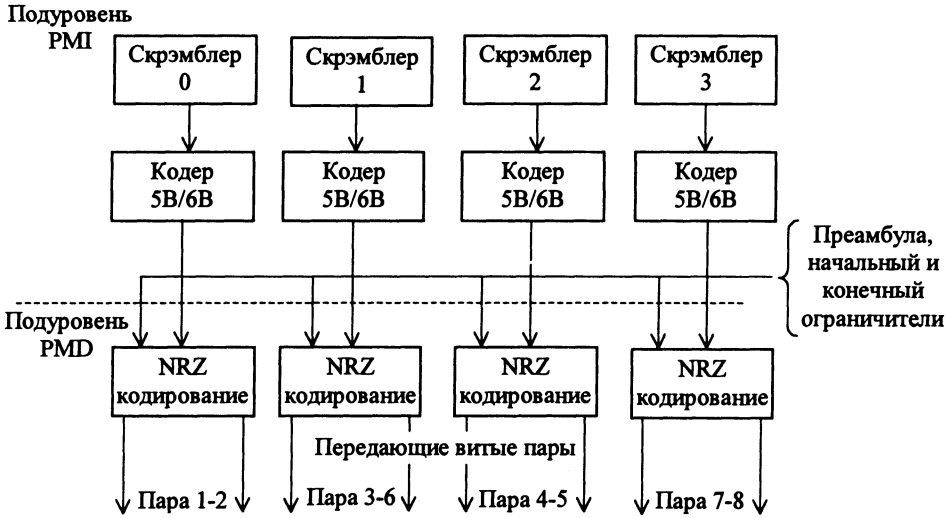


Рис. 3.13. Функции подуровней PMI и PMD

Функции подуровня PMI. Функции, не зависящие от физической среды, представленные на рис. 3.13, включают квартетную канальную шифрацию, кодирование 5B/6B, добавление к кадру преамбулы, начального и конечного ограничителей и передачу кадра на подуровень PMD.

Процесс квартетного распределения по каналам состоит в последовательном делении байтов MAC-кадра на порции данных по 5 бит (квинтеты), а также в последовательном распределении этих порций между четырьмя каналами, как это показано на рис. 3.14.

Каждый из 4 каналов представляет собой одну витую пару: канал 0 – пару, образованную контактами 1 и 2, канал 1 – пару 3 – 6; канал 2 – пару 4 – 5; канал 3 – пару 7 – 8. Двухпарные спецификации физического уровня PMD используют затем схему мультиплексирования, преобразующую 4 канала в 2 или 1.

Шифрация данных состоит в случайном «перемешивании» квинтетов данных с целью исключения комбинаций из повторяющихся единиц или нулей. Перемешивание производится с помощью специальных устройств – скрэмблеров. Случайные наборы цифр уменьшают излучение радиоволн и взаимные наводки в кабеле.

Кодирование по схеме 5B/6B – это процесс отображения «перемешанных» квинтетов в заранее определенные 6-битовые коды. Этот процесс создает сбалансированные коды, содержащие равное количество единиц и нулей, что обеспечивает гарантированную синхронизацию приемника при изменениях входного сигнала.

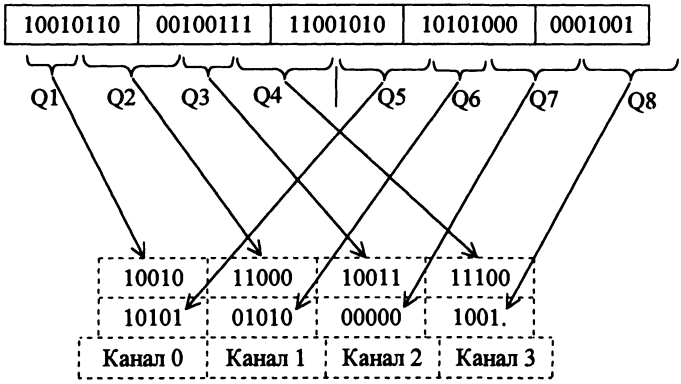


Рис. 3.14. Распределение квинтетов по каналам

Кодирование 5B/6B обеспечивает также контроль за ошибками при передаче, так как некорректные квинтеты, содержащие больше трех единиц или больше трех нулей, легко обнаружить.

Преамбула, начальный и конечный ограничители добавляются в каждом канале для корректной передачи данных через сеть.

Функции подуровня PMD. Функции зависимого от физической среды уровня PMD включают: мультиплексирование каналов (только для двух витых пар или оптоволокна), копирование NRZ, операции передачи сигналов по среде и контроль статуса физической связи.

Технология 100VG-AnyLAN поддерживает следующие типы физической среды:

- 4-парную неэкранированную витую пару;
- 2-парную неэкранированную витую пару;
- 2-парную экранированную витую пару;
- одномодовый или многомодовый оптоволоконный кабель.

Ниже будут рассмотрены детали спецификации PMD для 4-парной неэкранированной витой пары.

Спецификация 4UTP, использующая 4-парную неэкранированную витую пару, использует тактовый генератор с частотой 30 МГц для передачи данных со скоростью 30 Мбит/с по каждому из четырех каналов, что в сумме составляет 120 Мбит/с кодированных данных. Приемник получает кодированные данные со скоростью 30 Мбит/с по каждому каналу и преобразует их в поток исходных данных со скоростью 25 Мбит/с, что в результате дает пропускную способность в 100 Мбит/с. Такой метод представления данных в кабеле позволяет технологии 100VG-AnyLAN работать на голосовом кабеле (Voice-Grade) категории 3. Максимальная частота результирующего сигнала на кабеле не пре-

вышает 15 МГц, так как метод NRZ очень эффективен в отношении спектра сигналов. При тактовой частоте в 30 МГц частота 15 МГц генерируется только при передаче кодов 10101010, что является для спектра результирующего сигнала наихудшим случаем. При передаче других кодов частота сигнала будет ниже 15 МГц.

Операции передачи данных на 4-парном кабеле используют как полнодуплексный, так и полудуплексный режимы. Полнодуплексные операции применяют для одновременной передачи в двух направлениях (от узла к концентратору и от концентратора к узлу) сигнальной информации о состоянии линии. Сигнальная информация от концентратора передается по парам 1–2 и 3–6, а от узла – по парам 4–5 и 7–8.

Полудуплексные операции используются для передачи данных от концентратора узлу и от узла концентратору по всем четырем парам.

Сигнализация о статусе связи, осуществляемая в полнодуплексном режиме, использует два низкочастотных сигнала, обозначаемые тон 1 (*Tone* 1) и тон 2 (*Tone* 2). Тон 1 генерируется путем передачи с частотой 30 МГц по очереди кодов, состоящих из 16 единиц, и кодов, состоящих из 16 нулей. Результирующий сигнал имеет частоту около 0,9375 МГц. Тон 2 генерируется путем передачи с частотой 30 МГц по очереди кодов, состоящих из 8 единиц, и кодов, состоящих из 8 нулей. Результирующий сигнал имеет частоту примерно 1,875 МГц.

Взаимодействие между концентратором и узлом происходит путем параллельной передачи по двум парам комбинации из указанных двух тонов (табл. 3.6).

Таблица 3.6. **Комбинация тонов в стандарте 100VG-AnyLAN**

Комбинация тонов	Значение при приеме узлом	Значение комбинации при приеме концентратором
1 – 1	Простой (Idle)	Простой (Idle)
1 – 2	Поступление кадра	Запрос на передачу кадра с нормальным приоритетом
2 – 1	Зарезервировано	Запрос на передачу кадра с высоким приоритетом
2 – 2	Запрос на инициализацию процедуры подготовки линии	Запрос на инициализацию процедуры подготовки линии

Состояние простоя означает, что концентратор или узел не имеют кадров, ожидающих передачи. Состояние «поступление кадра» означает, что на данный порт может быть передан кадр. Узел должен прекратить передачу сигнальных тонов по каналам 2 и 3 для того, чтобы быть готовым принять кадр.

Рассмотрим последовательность событий в сети 100VG-AnyLAN при передаче кадра данных от одной станции другой через концентратор. Будем считать, что узел посылает в сеть один кадр данных с нормальным приоритетом. На рис. 3.15 приведены 7 этапов этого процесса.

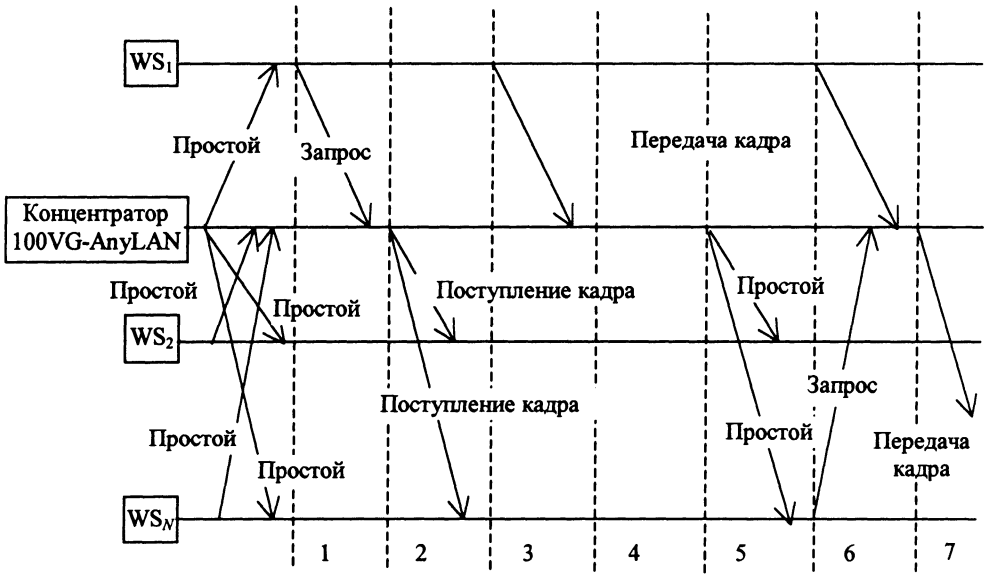


Рис. 3.15. Этапы передачи кадра данных через сеть 100VG-AnyLAN

Процесс начинается с получения MAC-уровнем конечного узла кадра данных от уровня LLC. После этого MAC-уровень добавляет к кадру адрес источника и дополняет поле данных, если сеть поддерживает формат кадров 802.3 и поле данных кадра LLC оказалось меньше 46 байт.

На этапе 1 узел WS_1 посылает в концентратор запрос нормального приоритета: тон 1 по каналу 2 и тон 2 по каналу 3. Во время цикла кругового опроса концентратор выбирает запрос узла WS_1 на обслуживание, в результате чего он прекращает генерацию комбинации сигнальных тонов «Простой» по каналам 0 и 1, очищая линию для передачи кадра по всем четырем каналам.

Концентратор предупреждает всех потенциальных получателей (узлы WS_2 – WS_N сети) о том, что им может быть направлен кадр данных. Для этого он посылает им сообщение «Поступление кадра» в форме тона 1 на канале 0 и тона 2 на канале 2 (этап 2). Узлы – потенциальные получатели кадра – прекращают посылку сигнальных тонов по каналам 2 и 3, очищая линию связи для передачи по всем четырем каналам кадра данных. Тем временем узел WS_1-источник кадра обнаруживает, что линия свободна и передает кадр с уровня MAC на уровень PMI для подготовки его к передаче по кабелю. Уровень PMI распределяет данные между четырьмя каналами, шифрует квинтеты данных и кодирует квинтеты в 60-битный код 5B/6B. При этом добавляются преамбула, стартовый и конечный ограничители по каждому каналу. Уровень PMD начинает передавать кадр концентратору, используя NRZ-кодирование (этап 3). По мере поступления данных кадра концентратор декодирует адрес назначения (этап 4).

Затем кадр передается через соответствующий порт узлу, который имеет адрес, совпадающий с адресом назначения кадра (этап 5, этап 7). В это же время концентратор перестает посылать сигнал «Поступление кадра» и начинает генерировать сигналы «Простой» всем остальным узлам. Эти узлы теперь могут посылать запросы на передачу своих кадров концентратору (этап 6).

Технология Token Ring

Сеть Token Ring разработана компанией IBM в 1970 г. Она по-прежнему является основной технологией IBM для локальных сетей. Фактически по образцу Token Ring IBM была создана спецификация IEEE 802.5, которая почти идентична и полностью совместима с сетью Token Ring. Термин «Token Ring» обычно применяется как при ссылке на сеть Token Ring IBM, так и на сеть IEEE 802.5.

Сети Token Ring и IEEE 802.5 являются примерами сетей с передачей маркера. Сети с передачей маркера перемещают вдоль сети небольшой блок данных, называемый маркером. Владение этим маркером гарантирует право передачи. Если узел, принимающий маркер, не имеет информации для отправки, он просто переправляет маркер к следующей конечной станции. Каждая станция может удерживать маркер в течение определенного времени.

Если у станции, владеющей маркером, есть информация для передачи, она захватывает маркер, изменяет у него один бит (в результате чего маркер превращается в последовательность «начало блока данных»), дополняет информацией, которую он хочет передать и, наконец, отсылает эту информацию к следующей станции кольцевой сети. Когда информационный блок циркулирует по кольцу, маркер в сети отсутствует (если только кольцо не обеспечивает «раннего освобождения маркера» – Early Token Release), поэтому другие станции, желающие передать информацию, вынуждены ожидать. Следовательно, в сетях Token Ring не может быть коллизий. Если обеспечивается раннее высвобождение маркера, то новый маркер может быть выпущен после завершения передачи блока данных.

Информационный блок циркулирует по кольцу, пока не достигнет предполагаемой станции назначения, которая копирует информацию для дальнейшей обработки. Информационный блок продолжает циркулировать по кольцу; он удаляется после достижения станции, отославшей этот блок. Станция отправки может проверить вернувшийся блок, чтобы убедиться, что он был просмотрен и затем скопирован станцией назначения.

В отличие от сетей CSMA/CD (например, Ethernet) сети с передачей маркера являются сетями с детерминированным методом доступа. Это означает, что можно вычислить максимальное время, которое пройдет, прежде чем любая конечная станция сможет передавать. Это предсказуемое значение максимального времени делает сеть Token Ring идеальной для применений, где задержка должна быть известна и важна устойчивость функционирования сети. Примерами таких применений является среда автоматизированных станций на заводах.

Технология Token Ring обеспечивает скорости передачи 4 Мбит/с или 16 Мбит/с.

В сетях Token Ring со скоростью передачи 16 Мбит/с используется алгоритм доступа к кольцу, называемый алгоритмом раннего освобождения маркера (Early Token Release). В соответствии с ним станция передает маркер доступа следующей станции сразу же после окончания передачи последнего бита кадра, не дожидаясь возвращения по кольцу этого кадра с битом подтверждения приема. В этом случае пропускная способность кольца используется более эффективно и приближается к 80 % от номинальной. Время удержания одной станцией маркера ограничивается тайм-аутом, после истечения которого станция обязана передать маркер далее по кольцу.

Не все станции в кольце равнозначны. Одна из станций обозначается как активный монитор, что означает дополнительную ответственность по управлению кольцом. Активный монитор осуществляет управление тайм-аутом в кольце, порождает новые маркеры (если необходимо), чтобы сохранить рабочее состояние, и генерирует диагностические кадры при определенных обстоятельствах. Активный монитор выбирается, когда кольцо инициализируется, и в этом качестве может выступить любая станция сети.

В сетях Token Ring используются три основных типа кадров (рис. 3.16):

Data/Command Frame (кадр управления/данные),

Token (маркер),

Abort (кадр сброса).

SD (Start Delimiter) – поле начального ограничителя. Оно появляется в начале маркера, и в начале любого кадра, проходящего по сети. Поле состоит из уникальной серии электрических импульсов, отличающихся от импульсов, которыми кодируются единицы и нули в байтах данных. Поэтому начальный ограничитель нельзя спутать ни с какой битовой последовательностью.

AC (Access Control) – поле управления доступом. Содержит поле приоритета P (3 бит), поле маркера T (1 бит), поле монитора M (1 бит) и резервное поле R (3 бит). Назначение этих полей P, T и M следующее.

• поле P (Priority) определяет уровень приоритета кольца: чем больше значение P, тем выше уровень приоритета;

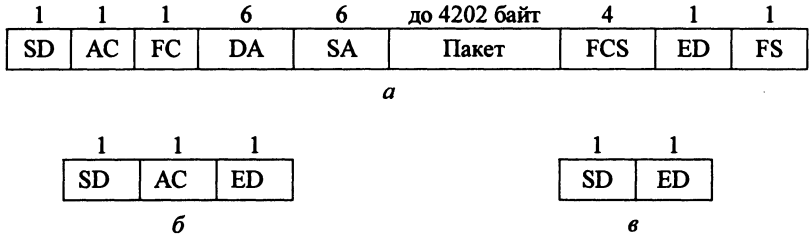


Рис. 3.16. Форматы кадров Token Ring:
цифры обозначают длины полей кадров в байтах

• поле T (Token). Значение поля равно 1, если это кадр Data/Command Frame, и 0, если это кадр Token;

• поле M (Monitor). Бит монитора устанавливается в «1» активным монитором и в «0» любой другой станцией, передающей маркер или кадр. Если активный монитор видит маркер или кадр, содержащий бит монитора в «1», то активный монитор знает, что этот кадр или маркер уже однажды обошел кольцо и не был обработан станциями. Если это кадр, то он удаляется из кольца. Если это маркер, то активный монитор переписывает приоритет из резервных битов полученного маркера в поле приоритета. Поэтому при следующем проходе маркера по кольцу его захватит станция, имеющая наивысший приоритет.

FC (Frame Control) – поле кадра управления. Для кадра управления в этом поле содержится команда управления. Это может быть команда инициализации кольца, команда проверки адресов устройств и т. п.

DA (Destination Address) – адрес приемника. Это может быть broadcast-multicast- или unicast-адрес.

SA (Source Address) – адрес источника.

Пакет – это данные, сформированные каким-либо протоколом (например, IPX). Максимальная длина пакета зависит от загрузки сети. При большой загрузке сети, когда многие станции имеют данные для передачи, интервал времени между получениями маркера станцией будет увеличиваться. В такой ситуации станции автоматически уменьшают максимальный размер пакета, поэтому каждая станция будет передавать свои данные за более короткий промежуток времени и, следовательно, уменьшится время получения (ожидания) маркера или время доступа станции к среде. Когда загрузка сети уменьшается, максимальный размер пакета динамически увеличивается. Этот механизм позволяет устойчиво работать сети Token Ring при пиковых нагрузках.

FCS (Frame Check Sequence) – контрольная сумма, вычисленная для полей FC, DA, SA, Пакет.

ED (End Delimiter) – конечный ограничитель кадра. Так же, как и поле начального ограничителя, это поле содержит уникальную серию электрических импульсов, которые нельзя спутать с данными. Кроме отметки конца маркера такое поле также содержит два подполя: один бит в этом подполе используется для индикации, что этот кадр является последним в логической цепочке, еще один бит изменяется приемником при обнаружении ошибки после сравнения контрольной суммы со значением в поле FCS.

FS (Frame Status) – поле статуса кадра. Данное поле состоит из полей A (Address Resolution) и C (Frame Copied). Передающая станция устанавливает эти поля в «0», а принимающая станция изменяет их в соответствии с результатами приема кадра и ретранслирует кадр дальше по сети. Когда кадр возвращается на станцию-передатчик, выполняется проверка полей A и C (табл. 3.7), и кадр удаляется из кольца.

Таблица 3.7. **Комбинация полей статуса кадра**

Поле		Описание
A	*C*	
0	0	Станция-приемник не доступна в данный момент
1	0	Станция-приемник обнаружила ошибку в кадре. Передача кадра повторяется
0	1	Недопустимая комбинация битов
1	1	Передача выполнена успешно

Рассмотрим механизм действия приоритетного маркерного кольца. Станция может воспользоваться кольцом, если только она получила маркер с приоритетом, меньшим или равным, чем ее собственный. Сетевой адаптер станции, если ему не удалось захватить маркер, помещает свой приоритет в резервные биты R маркера, но только в том случае, если записанный в резервных битах приоритет ниже его собственного. Эта станция будет иметь преимущественный доступ при последующем поступлении к ней маркера.

Предположим, что станции WS_1, WS_2 и WS_3 связаны в кольцо и имеют приоритеты 2, 5 и 4 соответственно. Сначала монитор помещает в поле текущего приоритета P максимальное значение приоритета P = 7, а поле резервного приоритета R обнуляется. Маркер проходит по кольцу, в котором станции имеют текущие приоритеты 2, 5 и 4. Так как эти значения меньше, чем 7, то захватить маркер станции не могут, но они записывают свое значение приоритета в поле резервного приоритета, если их приоритет выше его текущего значения. В результате маркер возвращается к монитору со значением резервного приоритета R = 5. Монитор переписывает это значение в поле P, а значение резервного приоритета обнуляет, и снова отправляет маркер по кольцу. При этом обороте его захватывает станция с приоритетом 5 – наивысшим приоритетом в кольце в данный момент времени. Передав данные, WS_2 сформирует и передаст кадр Token с приоритетом 5. Если ни одна станция в сети не имеет данных с таким приоритетом, то монитор сформирует маркер с полем P равным текущему значению резервного приоритета R (в нашем случае 4). Станция WS_3 захватит маркер и начнет передавать данные.

Пусть у станции WS_2 тоже появились данные для передачи. В это время через нее проходит кадр (данные от WS_3), где в поле AC установлен приоритет 4 (поле P). Тогда WS_2 запоминает старое значение P равное 4, и устанавливает в P значение своего приоритета, равное 5, т. е. присваивает кольцу более высокий уровень приоритета и ретранслирует кадр дальше. По кольцу этот кадр возвращается к станции-отправителю WS_3, она удаляет из кольца свой кадр и, обнаружив в поле P значение 5, формирует кадр Token (маркер) со значением поля P, равным 5, и направляет этот кадр по кольцу. Таким образом, станция WS_2 вновь получит право на передачу, поскольку у нее самый высокий приоритет. Передав данные, станция WS_2 «вспомнит», что в свое время увеличила приоритет, и восстановит его равным 4. Затем маркер с приоритетом 4 будет передан в сеть.

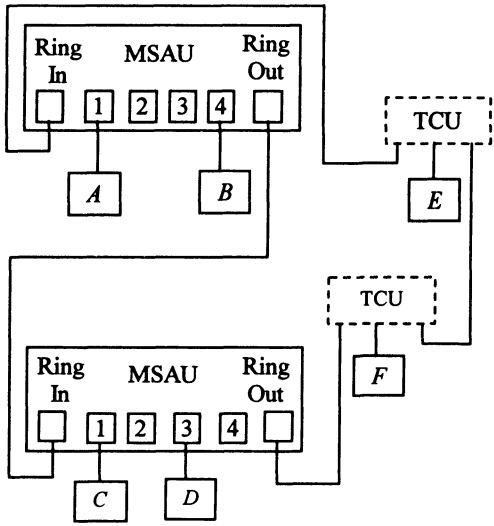


Рис. 3.17. Аппаратные элементы сети Token Ring

Кадр сброса состоит из двух байтов, содержащих начальный ограничитель и конечный ограничитель. Прерывающая последовательность может появиться в любом месте потока битов и сигнализирует о том, что текущая передача кадра или маркера отменяется.

В сети Token Ring на уровнях MAC и LLC применяются процедуры без установления связи, но с подтверждением получения кадров. Стандарт Token Ring фирмы IBM предусматривает построение связей в сети как с помощью непосредственного соединения станций друг с другом, так и образование кольца с помощью концентраторов (называемых MAU – Media Attachment Unit или MSAU – Multi-Station Access Unit). На рис. 3.17 показаны основные аппаратные элементы сети Token Ring и способы их соединения. Станции *A*, *B*, *C* и *D* подключаются к кольцу через концентраторы. Обычно такими станциями являются компьютеры с установленными в них сетевыми адаптерами. Станции этого типа соединяются с концентратором ответвительным кабелем (lobe cable), который обычно является экранированной витой парой (STP – Shielded Twisted Pair), соответствующей стандартному типу кабеля из кабельной системы IBM (Type 1, 2, 6, 8, 9). Максимальная длина ответвительного кабеля зависит от типа концентратора, кабеля и скорости передачи данных. Обычно для скорости 16 Мбит/с максимальная длина кабеля Type 1 может достигать 200 м, а для скорости 4 Мбит/с – 600 м. Концентраторы Token Ring делятся на активные и пассивные. Пассивные концентраторы обеспечивают только соединения портов внутри концентратора в кольцо, активные выполняют и функции повторителя, обеспечивая ресинхронизацию сигналов и исправление их амплитуды и формы. Естественно, что активные концентраторы поддерживают большие расстояния до станции, чем пассивные.

Станции *E* и *F*-сети соединены в кольцо непосредственными связями, называемыми магистральными (*trunk cable*), и используются для соединения концентраторов друг с другом для образования общего кольца. Порты концентраторов, предназначенные для такого соединения, называют портами Ring-In и Ring-Out.

Для предотвращения влияния отказавшей или отключенной станции на работу кольца, станции подключаются к магистрали кольца через специальные устройства, называемые устройствами подключения к магистрали (TCU – Trunk Coupling Unit). В функции такого устройства входит образование обходного пути, исключающего заход магистрали в MAC-узел станции при ее отключении или отказе. Обычно для этих целей в TCU используются реле, которые подпитываются постоянным током во время нормальной работы. При пропадании тока подпитки контакты реле переключаются и образуют обходной путь, исключая станцию.

При подключении станции в кольцо через концентратор, устройства TCU встраивают в порты концентратора. В одном кольце может быть до 250 станций.

Кроме экранированной витой пары существуют сетевые адаптеры и концентраторы Token Ring, поддерживающие неэкранированную витую пару и оптоволокно.

Технология сетей ARCNet

При подключении устройств в ARCNet применяют топологию «шина» или «звезда». Адаптеры ARCNet поддерживают метод доступа Token Bus (маркерная шина) и обеспечивают производительность 2,5 Мбит/с. Этот метод предусматривает следующие правила:

- все устройства, подключенные к сети, могут передавать данные, только получив разрешение на передачу (маркер);
- в любой момент времени только одна станция в сети обладает таким правом;
- кадр, передаваемый одной станцией, одновременно анализируется всеми остальными станциями сети.

В сетях ARCNet используется асинхронный метод передачи данных (в Ethernet и Token Ring – синхронный метод), т. е. передача каждого байта в них выполняется посылкой ISU (Information Symbol Unit – единица передачи информации), состоящей из трех служебных старт/стоповых битов и восьми битов данных.

В ARCNet определены 5 типов кадров (рис. 3.18 цифры обозначают длины полей кадров в байтах.):

- кадр ITT (Invitations To Transmit) – приглашение к передаче. Эта посылка передает управление от одного узла сети другому. Станция, принявшая такой кадр, получает право на передачу данных;
- кадр FBE (Free Buffer Enquiries) – запрос о готовности к приему данных. Этим кадром проверяется готовность узла к приему данных;
- кадр DATA – с его помощью передается пакет данных;

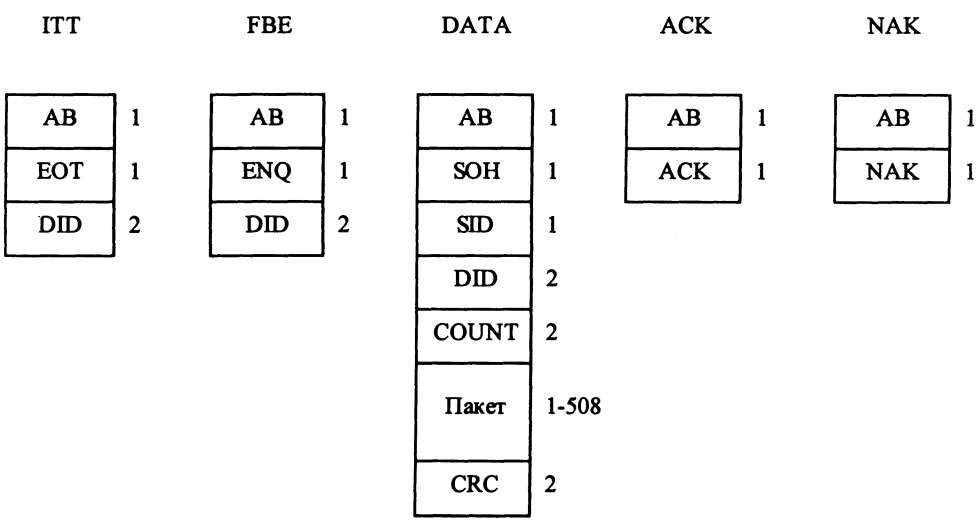


Рис. 3.18. Типы кадров для сетей ARCNet:
AB (Alert Burst) – начальный разделитель (выполняет функции преамбулы кадра); EOT (End Of Transmit) – символ конца передачи; DID (Destination Identification) – адрес приемника (ID-приемника). Если в поле заносится значение 00h, то кадр обрабатывается всеми станциями; ENQ (ENQuiry) – символ запроса о готовности к приему данных; SOH (Start Of Header) – символ начального заголовка; SID (Source Identification) – адрес источника (ID-источника); COUNT = 512–*N*, где *N* – длина пакета, байт; CRC – контрольная сумма; ACK (ACKnowledgments) – символ готовности к приему данных; NAK (Negative ACKnowledgments) – символ неготовности к приему данных

• кадр ACK (ACKnowledgments) – подтверждение приема. Подтверждение готовности к приему данных (ответ на FBE) или подтверждение приема кадра DATA без шибок (ответ на DATA);

• кадр NAK (Negative ACKnowledgments) – неготовность к приему. Неготовность узла к приему данных (ответ на FBE) или принят кадр с ошибкой (ответ на DATA).

Рассмотрим технологию сетей ARCNet на примере метода доступа Token Bus.

Все станции в сети ARCNet определяются 8-битовым ID (Identification – физический адрес сетевого адаптера). Этот адрес устанавливается переключателями на плате. Очередность передачи данных определяется физическими адресами станций (ID). Первой является станция с наибольшим адресом, затем следует станция с наименьшим адресом, далее – в порядке возрастания адресов. Каждая станция знает адрес следующей за ней станции (NextID или NID). Этот адрес определяется при выполнении процедуры реконфигурации системы. Выполнив передачу данных, станция передает право на передачу

данных следующей станции при помощи кадра ITT, при этом в поле DID устанавливается адрес NID. Следующая станция передает данные, затем кадр ITT и т. д. Таким образом, каждой станции предоставляется возможность передать свои данные. Предположим, что в сети работают станции с физическими адресами 3, 11, 14, 35, 126. Тогда маркер на передачу (кадр ITT) будет передаваться в следующей последовательности: 126→3→11→14→35→126→3 и т. д.

Для передачи пакета станция сначала должна получить маркер. Получив маркер, узел посылает кадр FBE той станции, которой должны быть переданы данные. Если станция-приемник не готова, она отвечает кадром NAK, в противном случае – ACK. Получив ACK, узел, владеющий маркером, начинает передавать кадр DATA. После отправки кадра передатчик ожидает ответ в течение 75,6 мкс. Если получен ответ ACK, то передатчик передает маркер следующей станции. Если получен ответ NAK, то передатчик повторно передает приемнику кадр DATA. Затем вне зависимости от ответа маркер передается следующей станции.

Каждая станция начинает принимать кадр DATA, обнаружив передачу начального разделителя AB. Затем она сравнивает значение адреса DID со своим адресом. Если адреса одинаковы или пришел broadcast-кадр, данные записываются в буфер станции, если нет – кадр игнорируется. Кадр считается нормально принятым, если он принят полностью и контрольная сумма совпадает со значением в поле CRC. Получив нормальный кадр DATA, станция передает ответ ACK. Если при приеме обнаружена ошибка, то передается ответ NAK. В ответ на широковещательный кадр DATA кадры ACK и NAK не передаются.

Рассмотрим теперь выполнение реконфигурации сети ARCNet. Реконфигурация сети выполняется автоматически всякий раз при включении новой станции или при потере маркера. Сетевой адаптер начинает реконфигурацию, если в течение 840 мс не получен кадр ITT. Это осуществляется посылкой специального кадра реконфигурации (Reconfiguration Burst). Такой кадр длиннее любого кадра, поэтому маркер будет разрушен (из-за коллизии) и никакая станция в сети не будет владеть маркером (т. е. правом на передачу). После приема кадра реконфигурации каждая станция переходит в состояние ожидания на время, равное 146×(256-ID) мкс. Если по окончании тайм-аута передач по сети не было (а это справедливо только для станции с наибольшим адресом ID), то узел передает кадр ITT с адресом DID, равным собственному ID. Если ни одна станция не ответила, узел увеличивает DID на единицу и повторяет передачу кадра ITT и т. д. После положительного ответа маркер передается ответившей станции, а ее адрес ID запоминается как адрес следующей станции (NID). Эта операция повторяется, пока маркер не вернется к первому узлу (станции с максимальным адресом). При выполнении реконфигурации каждая станция в сети узнает следующую за ней станцию. Таким образом формируется логическое кольцо, определяющее последовательность передачи маркера.

Технология FDDI

Сеть FDDI строится на основе двух оптоволоконных колец, образующих основной и резервный пути передачи данных между узлами сети. Использование двух колец – это основной способ повышения отказоустойчивости в сети FDDI. Узлы сети подключаются к обоим кольцам. В нормальном режиме работы сети данные проходят через все узлы и все участки кабеля первичного (Primary) кольца, поэтому этот режим назван режимом Thru – «сквозным» или «транзитным». Вторичное кольцо (Secondary) в этом режиме не используется.

В случае какого-либо вида отказа, когда часть первичного кольца не может передавать данные (например, обрыв кабеля или отказ узла), первичное кольцо объединяется со вторичным (рис. 3.19), образуя вновь единое кольцо. Этот режим работы сети называется Wrap, т. е. «свертывание» или «сворачивание» колец. Операция свертывания проводится концентраторами и/или сетевыми адаптерами FDDI. Для упрощения этой операции данные по первичному кольцу всегда передаются против часовой стрелки, а по вторичному – по часовой. 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.

В стандартах FDDI отводится много внимания различным процедурам, позволяющим определить наличие отказа в сети и провести необходимую реконфигурацию. Сеть FDDI может полностью восстанавливать свою работоспособность в случае единичных отказов ее элементов. При множественных отказах сеть распадается на несколько несвязанных сетей.

Кольца в сетях FDDI рассматриваются как общая разделяемая среда передачи данных, поэтому для нее определен специальный метод доступа. Этот метод очень близок к методу доступа сетей Token Ring и также называется методом маркерного кольца – token ring.

Механизм приоритета кадров в технологии FDDI отсутствует. Разработчиками технологии было принято решение о том, что деление трафика на 8 уровней приоритетов избыточно и достаточно разделить трафик на синхронный и асинхронный, первый из которых обслуживается всегда, даже при перегрузках кольца.

Рис. 3.19. Сворачивание колец FDDI в случае обрыва

Отличия метода доступа в сетях FDDI заключаются в том, что время удержания маркера не является постоянной величиной, как в сети Token Ring. Это время зависит от загрузки кольца: при небольшой загрузке оно увеличивается, а при больших перегрузках может уменьшиться до нуля. Такое изменение касается только асинхронного трафика, который допускает небольшие задержки передачи кадров.

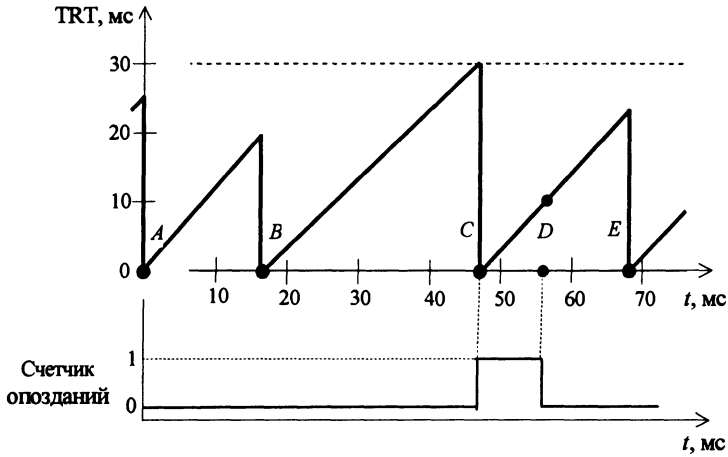


Рис.3.20. Поведение таймера времени текущего оборота маркера TRT и счетчика опозданий маркера Late Ct

Управление доступом к кольцу FDDI распределено между его станциями. Каждая станция при прохождении через нее маркера самостоятельно решает, может она его захватить или нет, а если да, то на какое время. Если у станции имеются для передачи синхронные кадры, то она всегда может захватить маркер на фиксированное время, выделенное ей администратором. Если же у станции имеются для передачи асинхронные кадры, то условия захвата определяются следующим образом. Станция ведет таймер текущего времени оборота маркера TRT (Token Rotation Timer), а также счетчик количества опозданий маркера Late Ct. Время отсчета таймера TRT равно значению максимального времени оборота маркера T Opr, выбранному станциями при инициализации кольца.

Счетчик Late Ct всегда сбрасывается в нуль, когда маркер проходит через станцию. Если же маркер опаздывает, то таймер достигает значения T Opr раньше очередного прибытия маркера. При этом таймер обнуляется и начинает отсчет времени заново, а счетчик Late Ct наращивается на единицу, фиксируя факт опоздания маркера. При прибытии опоздавшего маркера (при этом Late Ct = 1) таймер TRT не обнуляется, а продолжает считать, накапливая время опоздания маркера. Если же маркер прибыл раньше, чем истек интервал T Opr у таймера TRT, то таймер обнуляется в момент прибытия маркера.

На рис. 3.20 приведены различные случаи прибытия маркера. *A* – Маркер прибыл вовремя, так как таймер TRT не достиг порога T Opr; таймер TRT перезапускается и начинает считать заново. *B* – Маркер прибыл вовремя, таймер перезапускается. *C* – Таймер истек раньше, чем маркер прибыл на станцию; таймер TRT перезапускается, а счетчик опозданий Late Ct наращивается на единицу. *D* – Маркер прибыл, но опоздал – это отмечает счетчик опозданий Late Ct, равный «1»; счетчик сбрасывается в нуль, но таймер не перезапускается, так как при приходе маркера счетчик не был равен нулю. *E* – Маркер прибыл на станцию. Так как он прибыл до истечения таймера и при нулевом

значении счетчика опозданий Late Ct, то считается, что он прибыл вовремя; таймер перезапускается. Значение максимального времени оборота маркера для примера, приведенного на этом рисунке, равно 30 мс.

Рассмотрим, каким образом значения таймера TRT и счетчика Late Ct используются при выяснении возможности захвата маркера и времени его удержания. Станция может захватывать маркер только в том случае, когда он прибывает вовремя – т. е. в момент его прибытия счетчик Late Ct равен нулю. Время удержания маркера управляется таймером удержания маркера THT (Token Holding Timer). Если станция имеет в буфере кадры для передачи в момент прибытия маркера и маркер прибыл вовремя, то станция захватывает его и удерживает в течение периода T Opr – TRT (TRT – значение таймера TRT в момент прихода маркера). Для отслеживания разрешенного времени удержания маркера в момент захвата маркера значение TRT присваивается таймеру THT, а затем таймер TRT обнуляется и перезапускается. Таймер THT считает до границы T Opr, после чего считается, что время удержания маркера исчерпано. Станция перестает передавать кадры данных и передает маркер.

Описанный алгоритм позволяет адаптивно распределять пропускную способность кольца между станциями, точнее – ту ее часть, которая осталась после распределения между синхронным трафиком станций.

Пример работы алгоритма выделения времени для передачи асинхронного трафика приведен на рис. 3.21. Как и в предыдущем примере, время максимального оборота маркера равно 30 мс. *A* – Маркер прибыл вовремя, так как таймер TRT не достиг порога T Opr. Таймер TRT перезапускается и начинает считать заново. Станция не имеет в это время асинхронных кадров, поэтому просто передает маркер соседу. *B* – Маркер прибыл вовремя. Станция имеет к этому моменту асинхронные кадры для передачи. Таймеру THT присваивается значение таймера TRT (16), и он начинает считать до значения T Opr (30). Таймер TRT перезапускается. Станция начинает передавать кадры. Она может это делать в течение 14 мс. Если передача имеющихся кадров закончится раньше, то она обязана немедленно освободить маркер. *C* – Таймер THT истек, и станция должна прекратить передачу асинхронных кадров. Станция завершает передачу текущего кадра и передает маркер соседней станции. Счетчик TRT при этом продолжает работать. *D* – Таймер TRT истекает раньше очередного прибытия маркера. Таймер перезапускается, а счетчик опозданий Late Ct наращивается на 1. *E* – Маркер прибывает, но он опоздал, так как Late Ct имеет значение 1. Станция не может захватить маркер при значении Late Ct, отличном от нуля. Маркер передается соседней станции. Счетчик опозданий Late Ct обнуляется, а таймер TRT не перезапускается. *F* – Маркер прибывает на станцию. Так как таймер TRT еще не истек, а значение Late Ct равно «0», то маркер прибыл вовремя. Таймер THT инициализируется значением таймера TRT (22) и начинает считать до границы T Opr. Таймер TRT перезапускается. Станция может передавать кадры в течение 8 мс. *G* – Таймер THT истекает, и передача асинхронных кадров прекращается. Станция передает маркер соседней станции.

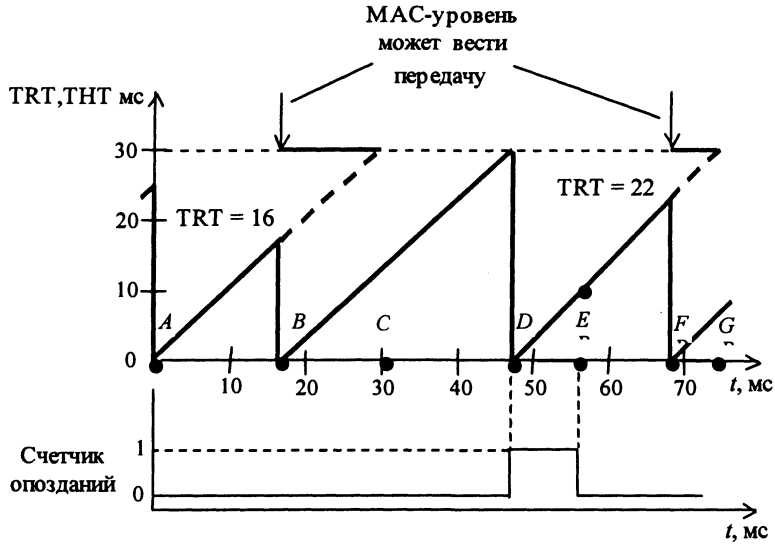


Рис. 3.21. Выделение времени для асинхронного трафика

В стандарте FDDI определено также еще два механизма управления доступом к кольцу. Во-первых, в маркере можно задавать уровень приоритета маркера, а для каждого уровня приоритета задается свое время порога, до которого считает таймер удержания маркера THT. Во-вторых, определена особая форма маркера – сдерживающий маркер (restricted token), с помощью которого две станции могут монопольно некоторое время обмениваться данными по кольцу.

Если таймер TRT истечет при значении Late Ct, равном 1, то такое событие считается потерей маркера и порождает выполнение процедуры инициализации кольца.

Процедура инициализации кольца, известная под названием Claim Token (в свободном переводе это можно интерпретировать как «соревнование претендентов на генерацию токена»), выполняется для того, чтобы все станции кольца убедились в его потенциальной работоспособности, а также пришли к соглашению о значении параметра TOpr – максимально допустимому времени оборота токена по кольцу, на основании которого все станции вычисляют время удержания токена THT.

Процедура Claim Token выполняется в нескольких ситуациях:

- при включении новой станции в кольцо и при выходе станции из кольца;
- при обнаружении какой-либо станцией факта утери токена. Токен считается утерянным, если станция не наблюдает его в течение двух периодов времени максимального оборота токена TOpr;
- при обнаружении длительного отсутствия активности в кольце, когда станция в течение определенного времени не наблюдает проходящих через нее кадров данных;
- по команде от блока управления станцией SMT.

Для выполнения процедуры инициализации каждая станция сети должна знать о своих требованиях к максимальному времени оборота токена по кольцу. Эти требования содержатся в параметре, называемом «требуемое время оборота токена» – TTRT (Target Token Rotation Time). Параметр TTRT отражает степень потребности станции в пропускной способности кольца – чем меньше время TTRT, тем чаще станции требуется токен для передачи своих кадров. Процедура инициализации позволяет станциям узнать о требованиях ко времени оборота токена других станций и выбрать минимальное время в качестве общего параметра T Opr, на основании которого в дальнейшем будет распределяться пропускная способность кольца. Параметр TTRT должен находиться в пределах 4...165 мс и может изменяться администратором сети.

Для проведения процедуры инициализации станции обмениваются служебными кадрами MAC-уровня – кадрами Claim. Эти кадры в поле управления имеют значение 1100 0011, поле адреса назначения содержит адрес источника (*DA = SA*), а в поле информации содержится 4-байтовое значение запрашиваемого времени оборота токена TReq.

Если какая-либо станция решает начать процесс инициализации кольца по своей инициативе, то она формирует кадр Claim Token со своим значением требуемого времени оборота токена TTRT, т. е. присваивает полю TReq свое значение TTRT. Захвата токена для отправки кадра Claim не требуется. Любая другая станция, получив кадр Claim Token, начинает выполнять процесс Claim Token. При этом станции устанавливают признак нахождения кольца в работоспособном состоянии Ring Operational в состояние False, что означает отмену нормальных операций по передаче токена и кадров данных. В этом состоянии станции обмениваются только служебными кадрами Claim.

Для выполнения процедуры инициализации каждая станция поддерживает таймер текущего времени оборота токена TRT, который используется также и в дальнейшем при работе кольца в нормальном режиме. Для упрощения будем считать, что этот таймер, как и другие таймеры станции, инициализируется нулевым значением и затем наращивает свое значение до определенного значения, называемого порогом истечения таймера. (В реальном кольце FDDI все таймеры работают в двоичном дополнительном коде.)

Таймер TRT запускается каждой станцией при обнаружении момента начала процедуры Claim Token. В качестве предельного значения таймера выбирается максимально допустимое время оборота токена, т. е. 165 мс. Истечение таймера TRT до завершения процедуры означает ее неудачное окончание – кольцо не удалось инициализировать. В случае неудачи процесса инициализации запускаются процессы Beacon и Trace, с помощью которых станции кольца пытаются выявить некорректно работающую часть кольца и отключить ее от сети.

Во время выполнения процесса инициализации каждая станция сначала может отправить по кольцу кадр Claim со значением TReq, равным значению ее параметра TTRT. При этом она устанавливает значение TOpr, равное значению TTRT.

Пример. Рассмотрим процесс инициализации кольца, приведенный на рис. 3.22. Пусть в некоторый момент времени все станции передали по кольцу свои предложения о значении мак-

симального времени оборота токена: 72, 37, 51 и 65 мс. Станция, приняв кадр Claim от предыдущей станции, обязана сравнить значение TReq, указанное в кадре со значением TTRT своего предложения. Если другая станция просит установить время оборота токена меньше, чем данная (т. е. TReq < TTRT), то данная станция перестает генерировать собственные кадры *Claim* и начинает повторять чужие кадры Claim, так как видит, что в кольце есть более требовательные станции. Одновременно станция фиксирует в своей переменной TOpr минимальное значение TReq, которое ей встретилось в чужих кадрах Claim. Если же пришедший кадр имеет значение TReq больше, чем собственное значение TTRT, то он удаляется из кольца.

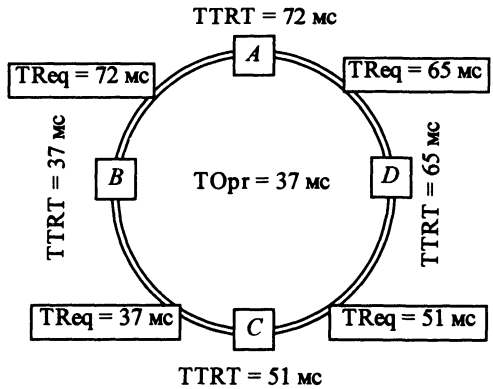


Рис. 3.22. Процесс инициализации кольца

Процесс Claim завершается для станции в том случае, если она получает кадр Claim со своим адресом назначения. Это означает, что данная станция является победителем состязательного процесса и ее значение TTRT оказалось минимальным. В рассматриваемом примере это станция B со значением TTRT, равным 37 мс. Другие станции кольца не смогут получить свой кадр Claim, так как он не сможет пройти через станцию B. При равных значениях параметра TTRT преимущество отдается станции с большим значением MAC-адреса.

После того, как станция обнаруживает, что она оказалась победителем процесса Claim Token, она должна сформировать токен и отправить его по кольцу. Первый оборот токена служебный, так как за время этого оборота станции кольца узнают, что процесс Claim Token успешно завершился. При этом они устанавливают признак Ring Operational в состояние True, означающее начало нормальной работы кольца. При следующем проходе токена его можно будет использовать для захвата и передачи кадров данных.

Если же у какой-либо станции во время выполнения процедур инициализации таймер TRT истек, а токен так и не появился на входе станции, то станция начинает процесс Beacon. После нормального завершения процесса инициализации у всех станций кольца устанавливается одинаковое значение переменной TOpr.

По сети FDDI информация передается в форме двух блоков данных: кадра и токена. Формат блоков данных FDDI представлен на рис. 3.23.

Рассмотрим назначение полей кадра:

• Преамбула (PA – Preamble). Любой кадр должен предваряться преамбулой, состоящей как минимум из 16 символов Idle (I). Эта последовательность

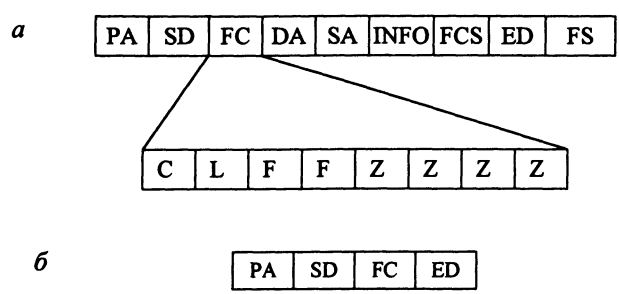


Рис. 3.23. Формат кадра и маркера FDDI

предназначена для вхождения в синхронизм генератора, обеспечивающего прием последующих символов кадра.

• Начальный ограничитель (SD – Starting Delimiter). Состоит из пары символов JK, позволяющих однозначно определить границы для остальных символов кадра.

• Поле управления (FC – Frame Control). Идентифицирует тип кадра и детали работы с ним, имеет 8-битовый формат и передается с помощью двух символов. Состоит из подполей, обозначаемых как CLFFZZZZ, которые имеют следующее назначение:

C – говорит о том, какой тип трафика переносит кадр – синхронный (значение 1) или асинхронный (значение 0);

L – определяет длину адреса кадра, который может состоять из 2 или 6 байт;

FF – тип кадра, может иметь значение 01 для обозначения кадра LLC (пользовательские данные) или 00 для обозначения служебного кадра MAC-уровня. Служебными кадрами MAC-уровня являются кадры трех типов: процедуры инициализации кольца Claim Frame, процедуры сигнализации о логической неисправности Beacon Frame и процедуры управления кольцом SMT Frame;

ZZZZ – детализирует тип кадра.

• Адрес назначения (DA – Destination Address). Идентифицирует станцию (уникальный адрес) или группу станций (групповой адрес), которой(ым) предназначен кадр. Может состоять из 2 или 6 байт.

• Адрес источника (SA – Source Address). Идентифицирует станцию, сгенерировавшую данный кадр. Поле должно быть той же длины, что и поле адреса назначения.

• Информация (INFO). Содержит информацию, относящуюся к операции, указанной в поле управления. Поле может иметь длину от 0 до 4478 байт (от 0 до 8956 символов). Стандарт FDDI допускает размещение в этом поле маршрутной информации алгоритма Source Routing, определенной в стандарте 802.5. При этом в два старших бита поля адреса источника SA помещается комбинация 102 – групповой адрес, комбинация, не имеющая смысла для адреса источника, а обозначающая присутствие маршрутной информации в поле данных.

• Контрольная последовательность (FCS – Frame Check Sequence). Содержит 32-битную последовательность, вычисленную по стандартному методу CRC-32, принятому и для других протоколов IEEE 802. Контрольная последовательность охватывает поля FC, DA, SA, INFO и FCS.

• Конечный ограничитель (ED – Ending Delimiter). Содержит единственный символ Terminate (T), обозначающий границу кадра. Однако за ним располагаются еще признаки статуса кадра.

• Статус кадра (FS – Frame Status). Первые три признака в поле статуса должны быть индикаторами ошибки (E – Error), распознавания адреса (Address recognized, A) и копирования кадра (C – frame Copied). Каждый из этих индикаторов кодируется одним символом, причем нулевое состояние индикатора обозначается символом Reset (R), а единичное – Set (S). Стандарт позволяет производителям оборудования добавлять свои индикаторы после трех обязательных.

Токен состоит по существу из одного значащего поля – поля управления, которое содержит в этом случае 1 в поле *C* и 0000 в поле *ZZZZ*.

Сеть FDDI кодирует информацию кодом 4B/5B, образуя символы. Символ – 5-битовая последовательность. Два символа составляют один байт. Это кодирование обеспечивает 16 символов данных (0–*F*), 8 контрольных символов (*Q*, *H*, *I*, *J*, *K*, *T*, *R*, *S*) и 8 символов нарушения (*V*), представленных в табл. 3.8.

Таблица 3.8. **Код 4B/5B**

Символ	Поток битов	Символ	Поток битов
0 (*binary* 0000)	11110	*Q*	00000
1 (*binary* 0001)	01001	*H*	00100
2 (*binary* 0010)	10100	*I*	11111
3 (*binary* 0011)	10101	*J*	11000
4 (*binary* 0100)	01010	*K*	10001
5 (*binary* 0101)	01011	*T*	01101
6 (*binary* 0110)	01110	*R*	00111
7 (*binary* 0111)	01111	*S*	11001
8 (*binary* 1000)	10010	*V* или *H*	00001
9 (*binary* 1001)	10011	*V* или *H*	00010
A (*binary* 1010)	10110	*V*	00011
B (*binary* 1011)	10111	*V*	00101
C (*binary* 1100)	11010	*V*	00110
D (*binary* 1101)	11011	*V* или *H*	01000
E (*binary* 1110)	11100	*V*	01100
F (*binary* 1111)	11101	*V* или *H*	10000

Технология виртуальных сетей

Технология виртуальных сетей (Virtual LAN) является одним из наиболее важных аспектов коммутируемых сетей, обеспечивая переход от сетей с разделяемой средой к полностью коммутируемым системам. Основное назначение виртуальных сетей – ограничить область распространения широковещательного трафика, т. е. организовать небольшие широковещательные домены. Виртуальные сети обеспечивают сегментацию за счет создания логических, динамических широковещательных доменов.

Подобно широковещательным доменам на базе маршрутизаторов в виртуальной ЛВС широковещательные пакеты и пакеты с неизвестными адресами получают все устройства, если такие пакеты происходят из того же домена (виртуальной сети). Здесь нет ничего нового, такие же методы используются в традиционных сетях на базе концентраторов и маршрутизаторов. Однако в традиционных сетях трафик является широковещательным внутри образующего сегмент концентратора и маршрутизируется между концентраторами. При ис-

пользовании виртуальных сетей кадры становятся широковещательными внутри VLAN и маршрутизируются между ними. Таким образом, виртуальные сети представляют собой не что иное, как более гибкий вариант традиционных ЛВС с несколько большими возможностями.

Виртуальная ЛВС (и связанные с ней коммутаторы) должна поддерживать различные типы физических сред. В коммутируемых сетях возможна работа централизованных ресурсов (магистралей) с более высокими скоростями, нежели скорость рабочих станций. Например, рабочие станции Ethernet (10 Мбит/с) могут работать с серверами Fast Ethernet, Gigabit Ethernet или ATM. Администратор сети должен быть уверен, что VLAN можно организовать для всех типов используемых в организации сетевых сред с учетом перспектив развития.

Каждый порт коммутатора должен обеспечивать поддержку более, чем одной виртуальной ЛВС. Это актуально даже в тех случаях, когда к портам коммутатора подключаются непосредственно рабочие станции (одной станции может потребоваться присутствие в нескольких виртуальных сетях). Некоторые коммутаторы ЛВС могут выполнять функции стандартной маршрутизации на сетевом уровне (IP и IPX), такая возможность позволяет организовать обмен данными между виртуальными ЛВС без использования внешних маршрутизаторов.

В эффективных реализациях виртуальных сетей серверы могут входить в несколько VLAN. Трафик в таком случае не передается через маршрутизатор или магистраль, что снижает нагрузку на сетевые магистрали и уменьшает задержку.

Во многих сетях устройства достаточно часто перемещаются с одного места на другое в пределах здания или территории предприятия. Администратор сети должен иметь возможность связать устройство или пользователя с виртуальными сетями независимо от местоположения. Использование коммутаторов обычно связано с необходимостью повышения производительности сети при одновременном снижении расходов на оборудование. Организация виртуальных ЛВС не должна снижать производительность сети. В виртуальных ЛВС широковещательный домен может объединять устройства, подключенные к одному или нескольким портам коммутатора или даже к портам разных коммутаторов. Так как VLAN организуются на базе логических групп пользователей, то расположение пользовательских станций не имеет значения в отличие от сетей на основе маршрутизаторов и концентраторов, где группы пользователей жестко определялись местоположением последних.

Для обмена между станциями в LAN значительные потоки данных передаются через маршрутизаторы. При организации логических групп потоки данных через маршрутизаторы можно уменьшить во много раз. Например, если пользователи сгруппированы в широковещательные домены так, что большая часть трафика остается внутри группы, нагрузка на магистрали и маршрутизаторы существенно снижается.

Широковещательный домен может содержать компьютеры, находящиеся в одном здании, городе или даже на значительном удалении друг от друга при поддержке виртуальных ЛВС с использованием WAN-каналов. Поскольку при обмене данными внутри группы маршрутизаторы не используются, обмен между станциями происходит гораздо быстрее.

Сегодня существует достаточно много вариантов реализации VLAN. Простые варианты VLAN представляют собой набор портов коммутатора, более сложные реализации позволяют создавать группы на основе других критериев. В общем случае возможности организации VLAN тесно связаны с возможностями коммутаторов.

Сети на базе портов. Это простейший вариант организации виртуальной ЛВС. VLAN на базе портов обеспечивают высочайший уровень управляемости и безопасности. Устройства связываются в виртуальные сети на основе портов коммутатора, к которым эти устройства физически подключены. VLAN на базе портов являются статическими и для внесения изменений необходимо физическое переключение устройств.

Однако построенные на базе портов виртуальные сети имеют некоторые ограничения. Они очень просты в установке, но позволяют поддерживать для каждого порта только одну виртуальную ЛВС. Следовательно, такое решение мало приемлемо при использовании концентраторов или в сетях с мощными серверами, к которым обращается много пользователей (сервер не удастся включить в разные VLAN). Кроме того, виртуальные сети на основе портов не позволяют вносить в сеть изменения достаточно простым путем, поскольку при каждом изменении требуется физическое переключение устройств.

Сети на базе MAC-адресов. Хотя этот тип виртуальных сетей относится к числу наиболее простых, VLAN на базе MAC-адресов настраивать сложнее, чем сети на основе физических портов. Виртуальная сеть на базе MAC-адресов группирует устройства, а программное обеспечение, например AutoTracker, делает группу широковещательным доменом (VLAN). Сети на базе MAC-адресов являются одним из наиболее безопасных и управляемых типов VLAN. Для получения доступа в виртуальную сеть устройство должно иметь MAC-адрес, известный программе AutoTracker.

Настройка виртуальной сети на основе MAC-адресов может отнять много времени. Кроме того, MAC-адреса «наглухо зашиты» в оборудование и может потребоваться много времени на выяснение адресов устройств в большой, территориально распределенной сети. Программа управления сетью OmniVision корпорации Xylan позволяет собрать адреса в масштабе всей сети автоматически, избавляя администратора от рутинной работы. С помощью этой программы можно настроить виртуальные сети, используя вместо MAC-адресов связанные с ними имена станций.

VLAN на сетевом уровне. Виртуальные ЛВС сетевого уровня позволяют администратору связать трафик для того или иного протокола в соответствующей виртуальной сети. Точно таким же способом создаются широковещатель-

ные домены в сетях на основе маршрутизаторов. Протокол может быть задан в форме IP-подсети или сетевого номера IPX. Можно, к примеру, объединить в виртуальную ЛВС всех пользователей подсети, которая была организована до использования коммутаторов.

Спектр возможностей коммутатора, на базе которого строится VLAN, определяет гибкость виртуальных сетей данного типа. Многие виртуальные ЛВС сетевого уровня поддерживают системы на базе нескольких коммутаторов, тогда как другие могут работать только с одним устройством.

VLAN на базе протоколов. Данный тип виртуальных сетей строится на базе заданного в каждом кадре типа протокола. Такой подход позволяет администратору задать критерии, по которым будет создаваться VLAN. Администратор может самостоятельно выбрать поля в заголовках кадров, по которым будет определяться принадлежность к виртуальной сети, и загрузить подготовленные правила во все коммутаторы сети. Таким образом, можно поместить в одну виртуальную сеть всех пользователей, работающих с протоколом NetBios или IP. Для работы с данным типом виртуальных сетей администратор должен досконально разбираться в заголовках широковещательных кадров.

После того, как правила загружены в коммутаторы, устройства начинают работу с виртуальными сетями на основе заданных администратором правил.

Многоадресные (multicast) VLAN. Многоадресный (multicast) трафик отличается от широковещательного (broadcast), который передается во всю сеть, и одноадресного (unicast), обеспечивающего связь «точка-точка». Многоадресный трафик представляет собой обмен «точка-многоточка» (один со многими) или многоточечный (многие со многими) и в последнее время становится все более популярным для различных сетевых приложений. Многоадресный режим можно использовать для видеоконференций, биржевых систем, новостей и подобных систем, где одна и та же информация передается многочисленным пользователям.

Виртуальные ЛВС с многоадресным трафиком создаются динамически путем прослушивания IGMP (Internet Group Management Protocol). Когда пользователь открывает приложение, использующее режим multicast, он динамически включается в виртуальную сеть, связанную с данным приложением. По окончании работы с программой пользователь удаляется из соответствующей виртуальной сети.

Многоадресный трафик в общем случае является стабильным потоком с достаточно широкой полосой. Следовательно, такой трафик лучше всего зафиксировать в одной виртуальной сети для предотвращения лавинной маршрутизации (flooding).

VLAN на базе правил. Это наиболее мощная реализация VLAN, позволяющая администратору использовать любые комбинации критериев для создания виртуальных ЛВС. Включение устройств в виртуальные ЛВС можно осуществить всеми перечисленными выше способами при условии их поддержки

установленными в сети коммутаторами. После того, как правила загружены во все коммутаторы, они обеспечивают организацию VLAN на основе заданных администратором критериев. Поскольку в таких сетях кадры постоянно просматриваются на предмет соответствия заданным критериям, принадлежность пользователей к виртуальным сетям может меняться в зависимости от текущей деятельности пользователей.

Виртуальные ЛВС на основе правил используют широкий набор критериев принадлежности к сети, включая все перечисленные выше варианты: MAC-адреса, адреса сетевого уровня, тип протокола и т.д. Можно также использовать любые комбинации критериев для создания правил, наиболее точно соответствующих вашим задачам.

VLAN для уполномоченных пользователей. VLAN для уполномоченных пользователей обеспечивают высокий уровень безопасности в сети и предъявляют более строгие требования к пользователям для предоставления доступа к серверам или иным сетевым ресурсам. Например, сеть уполномоченных пользователей может быть создана для финансового отдела предприятия, и сотрудники других подразделений не смогут получить доступ в эту сеть, не имея соответствующих полномочий. Для поддержки таких сетей в коммутаторах обычно используются функции встроенных брандмауэров. Администратор может эффективно управлять доступом пользователей, задавая процедуру аутентификации. Хотя и другие варианты VLAN обеспечивают некоторые средства безопасности, но только в сетях уполномоченных пользователей это выполняется на достаточно высоком уровне.

3.3. Оборудование локальных сетей

Структурированные кабельные системы

Основой любой компьютерной сети является среда для передачи данных, чаще всего это электрический кабель. Существует целая технология по разводке кабеля внутри здания или группы зданий. Крупные телефонные компании (такие, как AT&T) предлагают продукт, который так и называется – структурированная кабельная система (СКС). В это понятие помимо собственно конструктивных элементов входит также инженерная проработка проекта, включая настройку архитектуры сети на особенности здания и организации, которая будет ее использовать. Для тщательно спроектированной и построенной структурированной сети гарантируется, что в течение длительного периода (до 15 лет) даже появление новых технологий не потребует серьезных дорогостоящих изменений в кабельной системе.

Структурированная кабельная система представляет собой иерархическую кабельную систему здания или группы зданий, разделенную на структурные подсистемы. Она состоит из набора медных и оптических кабелей, коммутационных панелей или кросс-панелей (кроссов), соединительных шнуров, кабельных разъемов, модульных гнезд, информационных розеток и вспомогательного оборудования. Все элементы интегрируются в единую систему и эксплуатиру-

ются согласно определенным правилам. СКС обеспечивает подключение локальной АТС, одновременную работу компьютерной и телефонной сети, охранно-пожарной сигнализации, управление различными инженерными системами зданий и сооружений с использованием общей среды передачи. Эта система предоставляет также возможность гибкого изменения конфигурации кабельной сети. При перемещении служб и персонала внутри здания достаточно сделать необходимые переключения на коммутационных панелях.

Таким образом, СКС является универсальным и гибким решением задачи создания коммуникационной инфраструктуры здания или группы зданий. Как кабели являются основой сети, средой передачи данных, так коммутационные панели являются тем элементом, который обеспечивает такой сети гибкость и простоту конфигурирования. Кабельным разъемам в этой структуре отводится роль точки подключения сетевых устройств.

Топологически СКС представляет собой дерево, «листьями» которого являются кабельные разъемы на рабочих местах пользователей, узлами – коммутационные панели. Между собой все они соединены кабелем. Отличие лишь в том, что от кабельного разъема к коммутационной панели ведет один кабель; между собой, однако, панели соединяются несколькими кабелями. В коммутационных панелях обычно предусмотрена возможность соединения нескольких коммутаций в одном узле, в том числе и активного оборудования.

В коммутационных панелях, как правило, больше входных линий, чем выходных. Дело в том, что при построении сети розеток устанавливается больше, чем это необходимо. Розетка устанавливается не только на каждом рабочем месте, независимо от того, нужна она сегодня его владельцу или нет, но даже и там, где сегодня рабочего места нет, однако возможно его появление в будущем. Впоследствии переезд или подключение нового пользователя потребует лишь изменения коммутации на одной или нескольких панелях.

Как правило, панель первого уровня приходится одна на комнату, либо на этаж. Возможны и другие варианты, например, одна коммутационная панель обслуживает несколько этажей. В СКС есть и центральный коммутационный узел, куда сходятся кабели от панелей более низкого уровня и внешние коммуникации. Коммутационные панели принято устанавливать в помещения с контролируемым доступом, там же располагают и активное сетевое оборудование, а часто и сетевые серверы.

Существуют два варианта архитектуры проводки СКС:

- архитектура иерархической звезды;
- архитектура одноточечного управления.

Архитектура иерархической звезды применяется как для группы зданий, так и для одного отдельного здания. В первом случае иерархическая звезда состоит из центрального кросса системы, главных кроссов зданий и горизонтальных этажных кроссов. Центральный кросс связан с главными кроссами зданий внешними кабелями. Этажные кроссы связаны с главным кроссом здания кабелями вертикального ствола. Во втором случае звезда состоит из главного кросса здания и горизонтальных этажных кроссов, соединенных между собой кабелями вертикального ствола.

Архитектура иерархической звезды обеспечивает максимальные гибкость управления и способность адаптации системы к новым приложениям. Архитектура одноточечного администрирования разработана для наибольшей простоты управления. Обеспечивая прямое соединение всех рабочих мест с главным кроссом, она позволяет управлять системой из одной точки, оптимально расположенной для централизованного активного оборудования. Администрирование в одной точке обеспечивает простейшее управление цепями, возможное благодаря исключению необходимости кроссировки цепей во многих местах. Архитектура одноточечного администрирования не применяется для группы зданий.

Прорыв в области проектирования и создания СКС произошел в начале 90-х годов после принятия в июле 1991 г. в США стандарта EIA/TIA-568 (в октябре 1995 г. был принят новый стандарт EIA/TIA-568A) и выпуска сопутствующих документов. Позднее этот стандарт был дополнен документами TSB-36 (ноябрь 1991 г.) и TSB-40 (август 1992 г.), в которых определены категории 3, 4 и 5 для кабелей с неэкранированными витыми парами и соответствующего соединительного оборудования. Эти документы содержат технические требования к компонентам горизонтальной проводки, функционирующей на частотах до 100 МГц. Такая проводка поддерживает как давно существующие стандарты локальных сетей Ethernet и Token Ring, так и относительно недавно появившиеся Fast Ethernet, ATM, 100VG-AnyLAN.

В стандарте EIA/TIA-568 даны следующие рекомендации по составу и параметрам проводки:

- длина горизонтальных кабелей не должна превышать 90 м независимо от типа кабеля;
- к применению допускаются кабели четырех типов:
  - четырехпарный из неэкранированных витых пар с волновым сопротивлением 100 Ом;
  - двухпарный из экранированных витых пар с волновым сопротивлением 150 Ом;
  - коаксиальный (типа RJ-58) с волновым сопротивлением 50 Ом;
  - волоконно-оптический с волокнами диаметром 62,5/125 мкм;
- Следует использовать соответствующие соединители:
  - модульный восьмиконтактный RJ-45;
  - четырехконтактный, соответствующий стандарту IEEE 802,5;
  - коаксиальный BNC;
  - оптический (тип соединителя не определен);
- на каждом рабочем месте устанавливается модульная восьмиконтактная розетка типа RJ-45 или любая другая, соответствующая одному из перечисленных выше типов разъемов;
- разводка четырехпарного кабеля в соединителе RJ-45 должна быть выполнена по двум схемам:
  - TIA-568A;
  - TIA-568B (соответствует спецификации AT&T);

• разводка кабеля должна соответствовать топологии звезда.

В стандарте есть и другие рекомендации: о принципах размещения оборудования, способах соединения горизонтальной и вертикальной проводки и т. д.

Провода и кабели

Современные структурированные кабельные системы допускают использование следующих типов кабелей:

• коаксиальные;

• экранированные с витыми парами из медных проводников (STP – Shielded Twisted Pair);

• неэкранированные с витыми парами из медных проводников (UTP – Unshielded Twisted Pair);

• оптические (Fiber Optic Cable).

Коаксиальный кабель изготавливают двух видов: толстый и тонкий (thick и thin). Толстый кабель обеспечивает более надежную защиту от внешних шумов, он прочнее, передает информацию на значительные расстояния, но дорогой и требует использования специального отвода (прокалывающего разъема и отводящего кабеля) для подключения компьютера или другого устройства. Тонкий кабель (типа RJ-58) передает информацию на более короткие расстояния, однако он дешевле и для его подключения используют простые соединители.

Витая пара – это изолированные проводники, попарно свитые между собой минимально необходимое число раз на определенном отрезке длины, что необходимо для уменьшения перекрестных наводок между проводниками.

Для передачи информации по оптоволоконному кабелю используется свет. Оптоволоконный кабель позволяет передавать информацию на большие расстояния и с большой скоростью, однако этот кабель значительно дороже, сложнее в установке и обслуживании. Оптоволоконный кабель конструктивно несложен, но требует качественного монтажа. Он состоит из волокон диаметром в несколько микрон, окруженных твердым покрытием и помещенных в защитную оболочку. Первые оптоволоконные кабели изготовлялись из стекла, в настоящее время уже разработаны кабели на основе пластиковых волокон. Источником распространяемого по оптическим кабелям света является светодиод, а кодирование информации осуществляется изменением интенсивности света. На принимающем конце кабеля детектор преобразует световые сигналы в электрические. Различают два вида оптических кабелей: с одномодовым и многомодовым волокном. Одномодовый кабель может передавать данные на бóльшие расстояния, чем многомодовый, имеет меньший диаметр, однако он более дорогой. Исходя из соображений экономической эффективности и совместимости с основанным на оптике сетевым оборудованием, в большинстве случаев применяют многомодовое волокно. Одномодовое волокно следует использовать для передачи данных на большие расстояния (более 2 км) или, когда необходима очень высокая широкополосность.

Кабельные стандарты задают категорию проводки, необходимую для поддержки различных скоростей и расстояний. Совместный стандарт EIA/ TIA-568 определяет ряд параметров кабельных линий, от магистральной проводки (для связи телекоммуникационных шкафов с аппаратной внутри здания) до горизонтальной проводки (для подключения отдельных пользователей к стойке с сетевым оборудованием). В частности, стандарт дает рабочие характеристики магистрали, горизонтальных кабелей и типы соединителей, используемых с различными типами кабелей.

Стандарт EIA/TIA-568 допускает применение четырех типов кабелей. Наиболее популярным из них является UTP. Категории 3, 4 и 5 для неэкранированной витой пары поддерживают рабочие частоты до 16, 20 и 100 МГц соответственно. Согласно стандарту EIA/TIA-568, эти категории определяют поддерживаемый диапазон рабочих частот, а не скорость передачи данных по сети. Таким образом, скорость передачи сигналов конкретной локальной сети требуется сопоставить с аналогичной характеристикой для данной категории.

Каждая категория кабелей UTP, определенная EIA/TIA-568, имеет свои ограничения, выражаемые в терминах погонного и переходного затухания. Предельно допустимые значения погонного и переходного затухания для кабелей категории 3, 4 и 5 приведены в табл. 3.9. Так как категория 3 поддерживает частоты только до 16 МГц, а категория 4 – до 20 МГц, предельные значения на более высоких частотах для этих кабелей не указаны. Приведенные в таблице значения относятся только к определяемым стандартом расстояниям передачи. Для горизонтальной проводки максимальная протяженность кабеля между оборудованием в монтажном шкафу и информационной розеткой составляет 90 м. Соединительный кабель между информационной розеткой и адаптером локальной сети не должен превышать 10 м. Таким образом, общая длина подключения от компьютера до сетевого оборудования не должно превышать 100 м.

Предельные значения для погонного и переходного затухания определяются для каждой пары в кабеле; в этом разделе EIA/TIA не указано количество пар, необходимых для поддержки конкретного типа сети.

Таблица 3.9. **Предельные значения затухания и NEXT**

Рабочая частота, (МГц)	Категория кабеля					
	3		4		5	
	Предельные значения затухания и NEXT, дБ					
	Затухание	NEXT	Затухание	NEXT	Затухание	NEXT
10,0	11,5	22,7	7,5	36,6	7,0	44,0
20,0	–	–	11,0	31,4	10,3	39,0
100,0	–	–	–	–	24,0	27,1

Кабели с неэкранированными витыми парами должны удовлетворять определенным требованиям, сформулированным в согласованных между собой документах организаций IEEE, EIA/TIA (Electronic Industry Assotiation/ Telecommunication Industry Assotiation), NEMA (National Electrical Manu-facturers Assotiation) и др. Соответствие выпускаемых промышленностью кабелей

предъявляемым требованиям устанавливается в процессе сертификации. В США такого рода сертификацию проводит независимая организация UL, разработавшая специальную программу проведения испытаний и классификации кабелей по двум направлениям:

- по электробезопасности (требования сформулированы в американском стандарте NEC – National Electrical Code);
- по техническим характеристикам (требования сформулированы в документах EIA/TIA и NEMA).

Только после сертификационных испытаний фирма-изготовитель имеет право ставить на оболочке кабеля вместе со знаком UL соответствующее обозначение. Сертификация обеспечивает надежность и качество кабельной продукции на всех этапах изготовления, поставки и эксплуатации. В состав программы сертификации (Data Transmission Perfomance-Level Marking Program) входит определение уровня и маркировка кабелей из витых пар с волновым сопротивлением 100 Ом. Программа базируется на проверке выполнения требований промышленных стандартов для параметров и характеристик, описанных в спецификациях EIA/TIA, TSB-36 и др.

Стандарт NEC содержит самый полный набор требований по электробезопасности и включает, в частности, требования по пожаробезопасности кабелей. В соответствии с NEC наибольшую опасность при эксплуатации кабелей представляют инициирование огня электрическими цепями и распространение огня по кабелям. Требования пожаробезопасности определяются в зависимости от того, в каком виде проводки (горизонтальной или вертикальной) используются кабели. Кабели связи и слаботочные кабели должны быть отнесены к конкретному типу, определенному в стандарте NEC, в зависимости от использования внутри здания.

Оценку безопасности кабелей для локальных сетей выполняет UL по одному из двух стандартов: UL444 – для кабелей связи; UL13 – для слаботочных кабелей, причем кабели для локальных сетей можно оценить по любому из этих стандартов.

Программа сертификации по уровню характеристик проводится для кабелей связи и слаботочных кабелей. UL оценивает образцы кабелей по результатам испытаний всех характеристик, предусмотренных в сертификационной программе, а именно: сопротивление изоляции, асимметрия сопротивлений проводников, емкостная асимметрия, волновое сопротивление, структурные возвратные потери, коэффициент затухания и переходное затухание на ближнем конце.

Сертификации подвергаются не только кабели, но и процесс их производства. Чтобы поддерживать высокий уровень качества своей продукции, производители кабелей следуют одной из выбранных программ испытаний.

Классификация кабелей различных фирм-производителей по уровням, приведенная в табл. 3.10, соответствует стандарту качества ISO 9002. UL является участником программы ISO 9002 и осуществляет оценку качества процесса

производства. Сертифицированные кабели становятся объектом программы сопровождения (Follow Up Services Program), в соответствии с которой UL выполняет все последующие испытания, проводит необходимые инспекции на заводах-изготовителях, контролирует качество испытаний, проводимых изготовителями.

Таблица 3.10. **Классификация кабелей**

Рабочая полоса частот (скорость передачи)	Фирма производитель			
	Anixter	UL	EIA/TIA	AT&T
Передача речи и данных (до 20 кбит/с)	Level 1	Level I	–	–
1 МГц (1 Мбит/с)	Level 2	Level II	–	–
16 МГц (16 Мбит/с) 20 МГц (20 Мбит/с)	Level 3 Level 4	Level III Level IV	Category 3 Category 4	Category III Category IV
100 МГц (100 Мбит/с) 155 МГц (155 Мбит/с)	Level 5	Level V	Category 5	Category V

Follow-Up Services – необходимая часть сертификационных программ UL. Только постоянное сопровождение производства позволяет поддерживать высокое качество продукции и сохранять право маркировать изделия знаком UL.

Для подключения к толстому коаксиальному кабелю применяют специальное устройство – трансивер (transceiver). Трансивер снабжен коннектором, называемым «зуб вампира» (vampire tap) или «пронзающий ответвитель» (piercing tap). Этот «зуб» проникает через изоляционный слой и вступает в непосредственный электрический контакт с проводящей жилой. Чтобы подключить трансивер к сетевому адаптеру, надо кабель трансивера подключить к коннектору AUI-порта сетевой платы. Этот коннектор известен также как DIX-коннектор (Digital Intel Xerox), в соответствии с названиями фирм-разработчиков, или коннектор DB-15.

Для подключения тонкого коаксиального кабеля к компьютерам используют так называемые BNC-коннекторы (BNC – British Naval Connector). В семействе BNC представлено несколько компонентов:

BNC-коннектор либо припаивается, либо обжимается на конце кабеля. При помощи этого разъема кабель подключается к BNC Т-коннектору. Т-коннектор соединяет сетевой кабель с платой сетевого адаптера компьютера;

BNC-баррел-коннектор применяют для сращивания двух отрезков тонкого коаксиального кабеля;

BNC-терминатор используют для предотвращения отражения электрических сигналов в кабеле. Он устанавливается на каждом конце сегмента сети.

Для подключения витой пары к компьютеру используют телефонные коннекторы RJ-45. На первый взгляд, они похожи на RJ-11, но в действительности между ними есть существенные отличия. Во-первых, вилка RJ-45 чуть больше по размерам и не подходит для гнезда RJ-11. Во-вторых, коннектор RJ-45 имеет восемь контактов, а RJ-11 – только четыре.

Типы кроссовых панелей

Неотъемлемым элементом структурированных кабельных систем являются коммутационные или кроссовые панели (Cross Connect Panel), обеспечивающие коммутацию соединений кабелей горизонтальной и вертикальной проводки с портами активного сетевого оборудования (концентраторов, маршрутизаторов и т. д.). Существуют два основных типа кроссовых панелей. К первому типу относятся панели с врезными контактами. Они были разработаны телефонными компаниями для коммутации сотен и тысяч соединений, как правило, аналоговых. Контакты в этом соединителе относятся к соединителям со сдвигом изоляции (IDC – Insulation Displacement Connector). Лезвия контакта разрезают изоляцию провода при вставке, обеспечивая тем самым электрическое соединение с жилой провода и фиксацию провода в контакте.

Ко второму типу относятся модульные панели, разработанные специально для передачи данных. Эти панели имеют модульные гнезда для кабелей различных типов, например RJ-45 для UTP, BNC для тонкого коаксиального кабеля, ST или SC для оптоволоконного кабеля и т. д.

Панели с врезными контактами дешевле модульных и обеспечивают бóльшую гибкость и плотность соединений. Однако заделка проводов в них требует специальных инструментов и определенных навыков. Кроме того, существуют некоторые ограничения на число повторных заделок проводов в контакты с целью перекоммутации электрических цепей. Как правило, один и тот же контакт можно использовать не более 250 раз. Однако необходимость в таком числе перекоммутаций на практике возникает крайне редко. Для перекоммутации соединений на модульных панелях не нужно специальных навыков, и проводить ее можно до 750 раз с помощью стандартных соединительных шнуров.

Сетевые адаптеры

Сетевые адаптеры выступают в качестве физического интерфейса, или соединения, между компьютером и сетевым кабелем. Платы сетевых адаптеров вставляются в слоты расширения всех сетевых компьютеров и серверов. Чтобы обеспечить физическое соединение между компьютером и сетью, к соответствующему разъему, или порту, платы (после ее установки) подключается сетевой кабель.

При помощи сетевого адаптера осуществляется:

- преобразование данных, поступающих от компьютера, для их передачи по сетевому кабелю и обратное преобразование;
- передача данных другому компьютеру;
- управление потоком данных между компьютером и кабельной системой.

Плата сетевого адаптера состоит из аппаратной части и встроенных программ, записанных в ПЗУ. Эти программы реализуют функции подуровней управления логическим звеном (LLC) и управления доступом к среде (MAC) канального уровня модели OSI.

Плата сетевого адаптера принимает параллельные данные по внутренней шине компьютера и организует их для последовательной (serial), побитовой, передачи в сеть. Этот процесс завершается переводом цифровых данных компьютера в электрические и оптические сигналы, которые и передаются по сетевым кабелям. Осуществляет это преобразование приемо-передатчик, или трансивер.

Помимо преобразования данных, плата сетевого адаптера должна указать свое местонахождение, или адрес, чтобы ее могли отличить от остальных плат. Сетевые адреса или MAC-адреса определены комитетом IEEE. Этот комитет закрепляет за каждым производителем плат сетевого адаптера некоторый интервал адресов. Производители «зашивают» эти адреса в микросхемы, благодаря чему каждая плата и, следовательно, каждый компьютер имеют уникальный адрес в сети.

При приеме данных от компьютера и подготовке их к передаче по сетевому кабелю плата сетевого адаптера участвует также в других операциях:

- компьютер и плата сетевого адаптера должны быть связаны друг с другом, осуществлять передачу данных (от компьютера к плате). Если плата может использовать прямой доступ к памяти, компьютер выделяет ей некоторую область своей памяти;
- плата сетевого адаптера запрашивает у компьютера данные;
- шина компьютера передает данные из его памяти плате сетевого адаптера. Часто данные поступают быстрее, чем их способна передать плата сетевого адаптера.

Перед тем как послать данные по сети, плата сетевого адаптера проводит электронный диалог с принимающей платой, во время которого они «обговаривают»:

- максимальный размер блока передаваемых данных;
- объем данных, передаваемых без подтверждения о получении;
- интервалы между передачами блоков данных;
- интервал, в течение которого необходимо послать подтверждение;
- объем данных, который может принять каждая плата, не переполняясь;
- скорость передачи данных.

Если новой (более сложной и быстрой) плате необходимо взаимодействовать со старой (медленной) платой, они должны найти общую для обеих скорость передачи. Схемы некоторых современных плат сетевого адаптера позволяют им приспособиться к медленной скорости старых плат. Каждая плата оповещает другую о своих параметрах, принимая «чужие» параметры и подстраиваясь к ним. После того как все детали определены, платы начинают обмен данными.

Параметры платы сетевого адаптера должны быть корректно установлены, чтобы ее работа протекала правильно. В их число входят:

прерывание (IRQ);

базовый адрес порта ввода/вывода;

базовый адрес памяти;
используемый трансивер.

Линии запроса прерывания – это физические линии, по которым различные устройства (например, порты ввода/вывода, клавиатура, драйверы дисков и платы сетевого адаптера) могут послать микропроцессору компьютера запросы на обслуживание или на прерывание. Линии запроса прерывания встроены в аппаратуру компьютера и имеют различные уровни приоритетов, что позволяет процессору определить наиболее срочный из запросов.

Базовый адрес памяти (base address) указывает на ту область памяти компьютера, которая используется платой сетевого адаптера в качестве буфера для входящих и исходящих кадров данных. Этот адрес иногда называют начальным адресом. Часто базовым адресом памяти у платы сетевого адаптера является D8000. (Для некоторых плат последний нуль не указывается: вместо D8000 пишется D800). Необходимо выбирать базовый адрес памяти, не занятый другим устройством.

У плат сетевого адаптера, не использующих оперативную память, параметр базовый адрес памяти отсутствует.

Для обеспечения совместимости компьютера и сети, плата сетевого адаптера должна отвечать следующим требованиям;

- соответствовать внутренней структуре компьютера (архитектуре шины данных);
- иметь соединитель (он должен подходить к типу кабельной системы) для подключения сетевого кабеля.

К распространенным типам архитектуры шины данных относятся ISA, EISA, MCA-Channel и PCI. Каждая из них физически отличается от остальных. Плата сетевого адаптера должна соответствовать шине.

Координируя взаимодействие сетевого кабеля и компьютера, плата сетевого адаптера выполняет три важные функции:

организует физическое соединение с кабелем;
генерирует электрические сигналы, передаваемые по кабелю;
следует определенным правилам, регламентирующим доступ к сетевому кабелю.

Выбор платы сетевого адаптера определяется типом кабеля и разъемов, используемых в локальной сети. Каждый тип кабеля имеет различные физические характеристики, которым должна соответствовать плата. Сетевые адаптеры отличаются типом подключения (двойное или одиночное) и поддерживаемой средой передачи данных (оптоволокно или неэкранированная витая пара категории 5, тонкий или толстый коаксиальный кабель, витая пара или коаксиальный кабель).

Если у платы сетевого адаптера более одного интерфейсного разъема, выбор каждого из них осуществляют с помощью перемычек или DIP-переключателей, расположенных на самой плате, либо программно. Ниже приведены три примера типовых соединителей, которые можно найти на плате сетевого адаптера:

• для подключения тонкого коаксиального кабеля используют разъем для BNC-коннектора;

• для подключения толстого коаксиального кабеля применяют 15-контактный AUI-кабель, соединяющий 15-контактный (DB-15) разъем платы сетевого адаптера с внешним трансивером;

• для подключения витой пары применяют разъем RJ-45.

Поскольку плата сетевого адаптера оказывает существенное влияние на передачу данных, естественно, она влияет и на производительность всей сети. Если плата медленная, то и скорость передачи данных по сети не будет высокой. В сети с топологией «шина», где нельзя начать передачу, пока кабель занят, медленная сетевая плата увеличивает время ожидания для всех пользователей.

После определения физических требований к плате сетевого адаптера (выбор разъема и типа сети, в которой она будет использоваться) необходимо рассмотреть ряд факторов, влияющих на возможности платы.

Хотя все платы сетевого адаптера удовлетворяют определенным минимальным стандартам и спецификациям, некоторые из плат имеют дополнительные возможности, повышающие производительность сервера, клиента и всей сети.

К факторам, от которых зависит скорость передачи данных, относятся следующие.

Прямой доступ к памяти. Данные напрямую передаются из буфера платы сетевого адаптера в память компьютера, не затрагивая при этом центральный процессор.

Разделяемая память адаптера. Плата сетевого адаптера имеет собственную оперативную память, которую можно использовать совместно с компьютером. Компьютер воспринимает эту память как свою собственную.

Разделяемая системная память. Процессор платы сетевого адаптера использует для обработки данных память компьютера.

Управление шиной. К плате сетевого адаптера временно переходит управление шиной компьютера, минуя центральный процессор, плата передает данные непосредственно в системную память компьютера. При этом повышается производительность компьютера, так как его процессор в это время может решать другие задачи. Подобные платы дороги, но с их помощью можно повысить производительность сети на 20...70 %. Архитектуры EISA. MCA и PCI поддерживают этот метод управления.

Буферизация. Для большинства плат сетевого адаптера современные скорости передачи данных не слишком высоки. Поэтому на плате сетевого адаптера устанавливают буфер (с помощью микросхем памяти). В случае, когда плата принимает данных больше, чем способна обработать, буфер сохраняет их до тех пор, пока они не будут обработаны адаптером. Буфер повышает производительность платы, обеспечивая высокую скорость.

Встроенный микропроцессор. С таким микропроцессором плате сетевого адаптера для обработки данных не требуется помощь компьютера. Большинство сетевых плат имеет свои микропроцессоры, которые увеличивают скорость сетевых операций.

С серверами связана значительная часть сетевого трафика, поэтому их необходимо оборудовать платами сетевого адаптера с наибольшей производительностью. Рабочие станции могут использовать менее дорогие сетевые платы, если их работа с сетью ограничена приложениями, генерирующими небольшой объем сетевого трафика (например, текстовыми процессорами). Другие приложения (например, базы данных или инженерные приложения) довольно быстро перегрузят сетевые платы, не отвечающие их требованиям.

Бывают ситуации, когда безопасность данных настолько важна, что рабочие станции не оборудуются жесткими и гибкими дисками. Это гарантирует, что пользователи не смогут ни скопировать данные на какой-либо магнитный носитель, ни вынести диск с рабочего места.

Однако (поскольку обычно компьютер загружается с дискеты или с жесткого диска) необходимо иметь другой источник загрузки программного обеспечения, загружающего компьютер и подключающего его к сети. В таких ситуациях плата сетевого адаптера снабжается специальной микросхемой ПЗУ удаленной загрузки (PROM – remote-boot, которая содержит код для загрузки компьютера и для подключения его к сети и зависит от сетевой операционной системы). С такой микросхемой бездисковые рабочие станции при запуске могут подключаться к сети.

Большая часть производителей коммуникационного оборудования для локальных сетей поддерживают все современные технологии во всем спектре своих изделий: сетевых адаптерах, повторителях, коммутаторах и маршрутизаторах.

Концентраторы

Согласно классификационной базовой эталонной модели взаимодействия открытых систем сетевые (кабельные) концентраторы относятся к аппаратным средствам физического уровня передачи информации. Их основное назначение – коммутация линий связи, фильтрация, ретрансляция и усиление передаваемых сигналов в локальных сетях. Концентраторы, реализующие различные сетевые технологии, работают с различными видами носителей: коаксиальными кабелями, кабелями с экранированной и неэкранированной витой парой, волоконно-оптическими линиями связи.

Использование кабельных концентраторов позволяет увеличить количество абонентов сети и расстояние между ними. В некоторых системах (Token Ring) через концентраторы осуществляется доступ компьютеров к сети, в других их используют для изменения топологии сети и линий связи (10Base-T Ethernet).

Среди концентраторов выделяют *активные* (active), *пассивные* (passive) и гибридные (hybrid). Активные концентраторы регенерируют и передают сигналы так же, как это делают повторители сигналов или репитеры. Иногда их называют многопортовыми репитерами – они имеют от 8 до 12 портов для подключения компьютеров. Пассивные концентраторы (например монтажные панели или коммутирующие блоки) пропускают через себя сигнал как узлы коммутации, не усиливая и не восстанавливая его. Пассивные концентраторы не нужно подключать к источнику питания. Гибридными называют концентра-

торы, к которым можно подключать кабели различных типов. Сети, построенные на концентраторах, легко расширить, если подключить дополнительные концентраторы.

Использование концентраторов дает ряд преимуществ:

- разрыв кабеля в сети с обычной топологией «линейная шина» приводит к «падению» всей сети. В то время как разрыв кабеля, подключенного к концентратору, нарушает работу только данного сегмента. Остальные сегменты останутся работоспособными;
- простота изменения или расширения сети: достаточно просто подключить еще один компьютер или концентратор;
- использование различных портов для подключения кабелей разных типов;
- централизованный контроль за работой сети и сетевым трафиком: во многих сетях активные концентраторы наделены диагностическими возможностями, позволяющими определять работоспособность соединения.

Конструктивно можно выделить два типа концентраторов:

модульные устройства, выполненные в виде приборного блока, в посадочные места которого включают от 4 до 16 плат различного функционального назначения;

функционально законченные автономные устройства, предназначенные, как правило, для работы с одним типом сети и конкретным видом линий связи.

В состав модульных концентраторов могут входить специализированные платы, поддерживающие широкий спектр сетевых функций (маршрутизаторы, мосты, терминальные серверы, преобразователи протоколов) и узлы, резервирующие выполнение функций всех уровней, включая источники питания и охлаждающие вентиляторы. Замена одного типа модуля на другой или отказавшего устройства на резервное обычно проводится без выключения концентратора.

Среди относительно недорогих автономных концентраторов существует отдельный класс изделий – наращиваемые концентраторы. Автономные концентраторы рассчитаны на включение в сеть ПК с использованием какого-либо одного типа кабеля – витой пары, волоконно-оптического и др. Объединение нескольких наращиваемых концентраторов (обычно до 4), осуществляемое с помощью разъемов в корпусе устройства и простых кабельных соединений, позволяет создать сетевое устройство с единой системой управления портами, поддерживающее передачу информации по линиям связи различного типа. Интеграция функциональных возможностей концентратора дает дополнительные преимущества. Например, путем установки модуля в одном из них можно управлять всеми портами, изменять конфигурацию сети и предоставлять информацию, связанную со статистикой ее работы и состоянием устройств.

Применение наращиваемых концентраторов является экономичным решением, позволяющим создавать сети небольших организаций, подразделений и рабочих групп. При относительно невысоких начальных затратах обеспечивается гибкость, постепенное увеличение функциональных возможностей и интеграция отдельных сетевых фрагментов в единую сеть масштаба предприятия.

Модульные многофункциональные устройства называют концентраторами, чтобы подчеркнуть их централизующую роль в сети. При этом термин «концентратор» используют не как синоним термина повторитель, а в более широком смысле. Нужно хорошо понимать в каждом конкретном случае функциональное назначение отдельных модулей такого концентратора. В зависимости от комплектации модульный многофункциональный концентратор может сочетать функции и повторителя (причем различных технологий), и моста, и коммутатора, и маршрутизатора, а может выполнять и только одну из них.

Модульные концентраторы устанавливают в высокоскоростных локальных сетях с централизованным управлением. Кабельные концентраторы различных изготовителей могут сильно отличаться. Разработчики создают все более модульные и расширяемые системы, постоянно совершенствуя свою продукцию и оставляя задел для будущего роста. Перечислим некоторые возможности концентраторов:

- дополнительные источники питания. В некоторых модулях предусмотрены резервные источники питания, подключаемые при сбое в основном питании;
- оперативно заменяемые модули. Это средство позволяет заменить модуль, не прекращая работы всего блока;
- управление. Средства управления обычно имеют возможность удаленного управления и поддержку SNMP;
- простая настройка конфигурации. Когда рабочие станции или пользователи перемещаются из одной рабочей группы в другую, концентратор должен обеспечивать простое изменение конфигурации. Некоторые системы позволяют делать это программно.

Мосты и коммутаторы

По мере расширения сети доступная пользователю полоса (средняя скорость передачи) сужается за счет того, что моноканал делится между всеми узлами сети. Для полной реализации возможностей программ, оборудования, повышения производительности компьютеров, использующих приложения с интенсивным сетевым трафиком необходимо расширение полосы пропускания канала.

Существует два способа расширения полосы, доступной каждому пользователю. Первый базируется на расширении полосы разделяемой среды, обеспечивая рост скорости, например технология Fast Ethernet. Другим способом является снижение числа узлов сети, имеющих доступ к разделяемой среде и, следовательно, расширение доступной оставшимся узлам полосы. В предельном случае вся полоса канала передачи может быть предоставлена одному пользователю.

Процесс снижения числа узлов в сети называется сегментацией и осуществляется за счет деления большой сети на несколько меньших. Поскольку пользователям может потребоваться доступ к ресурсам других сегментов, то необходим механизм обеспечения такого доступа, обеспечивающий межсегментный обмен с достаточно высокой скоростью.

Как известно, основной задачей повторителя является восстановление электрических сигналов для передачи их в другие сегменты. За счет усиления и восстановления формы электрических сигналов повторителем возможно расширение сетей, построенных на основе коаксиального кабеля и увеличение общего числа пользователей сети.

Мосты (bridge) имеют много отличий от повторителей. Повторители передают все пакеты, а мосты только те, которые необходимы. Если пакет не нужно передавать в другой сегмент, он фильтруется. Для мостов существуют многочисленные алгоритмы (правила) передачи и фильтрации пакетов. Минимальным требованием является фильтрация пакетов по MAC-адресу получателя.

Другим важным отличием мостов от повторителей является то, что сегменты, подключенные к повторителю, образуют одну разделяемую среду, а сегменты, подключенные к каждому порту моста, образуют свою среду. Следовательно, мост обеспечивает преимущества как с точки зрения расширения сети, так и обеспечения бóльшей полосы для каждого пользователя.

В первых сетях Ethernet использовалась шинная топология на основе коаксиального кабеля, а для расширения сетей применялись 2-портовые повторители или мосты. Технология 10Base-T привела к трансформации топологии сетей от шинной магистрали к организации соединений типа «звезда». Требования к повторителям и мостам для таких сетей существенно изменились по сравнению с простыми двухпортовыми устройствами для сетей с шинной топологией. Современные мосты и повторители представляют собой сложные многопортовые устройства. Мосты позволяют сегментировать сети на меньшие части, в которых общую среду разделяет небольшое число пользователей.

Маршрутизаторы, подобно мостам, также позволяют сегментировать сети на сетевом уровне, фильтруя и пересылая сетевой трафик на основе алгоритмов и правил, существенно отличающихся от тех, что используются мостами.

Сегодняшние модульные концентраторы (повторители) обеспечивают организацию нескольких сегментов, каждый из которых предоставляет пользователям отдельную разделяемую среду. Некоторые концентраторы разрешают программным путем разделять порты устройства на независимые сегменты, реализуя тем самым функции моста. Такая возможность называется переключением портов. Переключение портов обеспечивает администратору сети высокую гибкость организации сегментов, позволяя переносить порты из одного сегмента в другой программными средствами. Эта возможность особенно полезна для распределения нагрузки между сегментами сети и снижения расходов, связанных с подобными операциями.

Коммутатор (Switch) подобно мостам и маршрутизаторам позволяет сегментировать сети. Как и многопортовые мосты, коммутаторы передают пакеты между портами на основе MAC-адреса получателя, включенного в каждый пакет. Реализация коммутаторов обычно отличается от мостов в части возможности организации одновременных соединений между любыми парами пор-

Таблица 3.11. **Внутренняя таблица коммутатора**

MAC-адрес	Номер порта
A	1
B	2
C	3
D	4

тов устройства, что значительно расширяет суммарную пропускную способность сети. Более того, мосты в соответствии со стандартом IEEE 802.1d должны получить пакет целиком до того, как он будет передан адресату, а коммутаторы могут начать передачу пакета, не приняв его полностью.

Коммутатор поддерживает внутреннюю таблицу, связывающую порты с адресами подключенных к ним устройств (табл. 3.11). Эту таблицу администратор сети может создать самостоятельно или задать ее автоматическое создание средствами коммутатора. Используя табл. 3.11 и содержащийся в пакете адрес получателя, коммутатор организует виртуальное соединение порта отправителя с портом получателя и передает пакет через это соединение. Например, узел A посылает пакет узлу D. Найдя адрес получателя в своей внутренней таблице, коммутатор передает пакет в порт 4.

Виртуальное соединение между портами коммутатора сохраняется в течение передачи одного пакета, т.е. для каждого пакета виртуальное соединение организуется заново на основе содержащегося в этом пакете адреса получателя.

Поскольку пакет передается только в тот порт, к которому подключен адресат, остальные пользователи (в нашем примере – B и C) не получат этот пакет. Таким образом, коммутаторы обеспечивают средства безопасности, недоступные для стандартных повторителей и концентраторов.

В коммутаторах передача данных между любыми парами портов происходит независимо и, следовательно, для каждого виртуального соединения выделяется вся полоса канала. Например, коммутатор 10 Мбит/с обеспечивает одновременную передачу пакета из A в D и из порта B в порт C с полосой 10 Мбит/с для каждого соединения. Поскольку для каждого соединения предоставляется полоса 10 Мбит/с, суммарная пропускная способность коммутатора в приведенном примере составляет 20 Мбит/с. Если данные передаются между большим числом пар портов, интегральная полоса соответственно расширяется. Теоретически, интегральная полоса коммутатора растет пропорционально числу портов. Однако в реальности скорость пересылки пакетов, измеренная в Мбит/с, меньше чем суммарная полоса пар портов за счет так называемой внутренней блокировки. Для коммутаторов высокого класса блокировка весьма незначительно снижает интегральную полосу устройства.

Коммутатор может обеспечить высокую пропускную способность при условии организации одновременных соединений между всеми парами портов. Однако реально трафик обычно представляет собой ситуацию «один ко многим» (например, множество пользователей сети обращается к ресурсам одного сервера). В таких случаях коммутатор не обеспечивает существенного преимущества по сравнению с обычным концентратором (повторителем).

В зависимости от назначения все коммутаторы имеют целесообразно разделить на три класса:

настольные коммутаторы;

коммутаторы рабочих групп;

магистральные коммутаторы.

Настольные коммутаторы предназначены для работы с небольшим числом пользователей и могут служить для замены концентраторов 10Base-T. Обычно настольные коммутаторы имеют 24 порта, каждый из которых поддерживает персональный (private) канал с полосой 10 Мбит/с для подключения одного узла (например, рабочей станции). Дополнительно такой коммутатор может иметь один или несколько портов 100Base-T или FDDI для подключения к магистрали (backbone) или серверу. Поддерживая на каждый порт по крайней мере то число адресов, которые могут присутствовать в сегменте, коммутатор обеспечивает для каждого порта выделенную полосу. Каждый порт коммутатора связан с уникальным адресом подключенного к данному порту устройства Ethernet.

Объединяя в себе возможности технологий 10 и 100 Мбит/с, настольные коммутаторы минимизируют блокировку при попытке одновременного подключения нескольких узлов к единственному скоростному порту (100 Мбит/с). В среде клиент-сервер одновременно несколько узлов могут получить доступ к серверу, подключенному через порт 100 Мбит/с. Например, если три узла 10 Мбит/с одновременно обращаются к серверу через порт 100 Мбит/с, то из полосы 100 Мбит/с, доступно серверу, используется 30 Мбит/с, а 70 Мбит/с доступно для одновременного подключения к серверу еще семи устройств 10 Мбит/с через виртуальные каналы.

Настольные коммутаторы просты в установке и обслуживании, зачастую содержат встроенные plug-and-play программы и имеют упрощенный интерфейс установки параметров. Стоимость в пересчете на один порт менее чем вдвое превосходит стоимость порта в концентраторах 10Base-T.

Коммутаторы рабочих групп используют главным образом для соединения изолированных настольных коммутаторов или концентраторов 10Base-T с другими частями сети. Эти устройства объединяют в себе свойства как настольных, так и магистральных коммутаторов.

Коммутаторы рабочих групп могут поддерживать множественную адресацию (до нескольких тысяч MAC-адресов на коммутатор), могут выступать в качестве маршрутизаторов, а также обеспечивают подключение к порту отдельных узлов.

Соединение со скоростью 10 Мбит/с между коммутатором и пользовательским узлом (рабочей станцией) чаще всего выполняется кабелем на основе UTP, а для скоростного порта используют UTP или оптический кабель. Групповые коммутаторы могут поддерживать несколько тысяч MAC-адресов на устройство.

Основным преимуществом коммутаторов рабочих групп является высокая производительность сети на уровне рабочей группы за счет предоставления каждому пользователю выделенной полосы канала (10 Мбит/с). Кроме того, это обьясняется тем, что коммутаторы рабочих групп, не передают коллизионные фрагменты адресатам и, следовательно, снижают (в пределе до нуля) количество коллизий. Коммутаторы рабочих групп позволяют полностью сохранить сетевую инфраструктуру со стороны клиентов, включая программы, сетевые адаптеры, кабели. Их стоимость в расчете на один порт сегодня сравнима с ценами портов управляемых концентраторов.

Магистральный коммутатор обеспечивает одновременную передачу пакетов со скоростью среды между любыми парами своих портов. Если скорость портов для отправителя и получателя совпадают, сегмент получателя должен быть свободен во избежание блокировки.

Магистральные коммутаторы находятся на вершине иерархии коммутаторов. Их используют для соединения сетей или сегментов, так как они поддерживают множественную адресацию для своих портов, а также для соединения концентраторов 10Base-T, настольных и групповых коммутаторов, серверов. Кроме того, магистральные коммутаторы могут фильтровать пакеты на основе признаков, отличающихся от адресов. Например, администратор может запретить передачу широковещательных пакетов NetWare рабочим станциям *Unix* за счет фильтрации по протоколу.

Для магистральных коммутаторов характерно модульное устройство и способность поддерживать до нескольких тысяч MAC-адресов на каждый порт.

Подобно коммутаторам рабочих групп, магистральные коммутаторы поддерживают различную скорость для своих портов. Они могут работать с сегментами 10Base-T и сегментами на основе коаксиального кабеля. В большинстве случаев использование магистральных коммутаторов обеспечивает более простой и эффективный способ повышения производительности сети по сравнению с маршрутизаторами и мостами.

Основным недостатком работы магистральных коммутаторов является то, что на уровне рабочих групп пользователи работают с разделяемой средой, если они подключены к сегментам, организованным на основе повторителей или коаксиального кабеля. Более того, время отклика на уровне рабочей группы может быть достаточно большим. В отличие от узлов, подключенных к портам коммутатора, для узлов, находящихся в сегментах 10Base-T или сегментах на основе коаксиального кабеля (10Base-2 или 10Base-5) полоса 10 Мбит/с не гарантируется, и они зачастую вынуждены ждать, пока другие узлы не закончат передачу своих пакетов. На уровне рабочей группы по прежнему сохраняются коллизии, а фрагменты пакетов с ошибками будут пересылаться во все сети, подключенные к магистрали. Перечисленных недостатков можно избежать, если на уровне рабочих групп использовать коммутаторы (вместо концентраторов) 10Base-T. В большинстве ресурсоемких приложений коммутатор 100 Мбит/с может выполнять роль скоростной магистрали для коммутаторов рабочих групп с портами 10 и 100 Мбит/с, концентраторами 100 Мбит/с и серверами, в которых установлены адаптеры Ethernet 100 Мбит/с.

Основными характеристиками коммутатора, измеряющими его производительность, являются:

- скорость фильтрации (filtering);
- скорость продвижения (forwarding);
- пропускная способность (throughput);
- задержка передачи кадра.

На указанные характеристики производительности влияют: размер буфера (буферов) кадров, производительность внутренней шины, производительность процессора или процессоров, размер внутренней адресной таблицы коммутатора.

Скорость фильтрации и продвижения кадров – две основные характеристики производительности коммутатора. Эти характеристики являются интегральными показателями, они не зависят от того, каким образом технически реализован коммутатор.

Скорость фильтрации определяет скорость, с которой коммутатор выполняет следующие этапы обработки кадров:

прием кадра в свой буфер;

просмотр адресной таблицы с целью нахождения порта для адреса назначения кадра;

уничтожение кадра, так как его порт назначения совпадает с портом-источником.

Скорость продвижения определяет скорость, с которой коммутатор выполняет следующие этапы обработки кадров:

прием кадра в свой буфер;

просмотр адресной таблицы с целью нахождения порта для адреса назначения кадра;

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 размера кадра приведены значения скоростей фильтрации и продвижения, то по умолчанию считается, что эти показатели даются для протокола Ethernet и кадров минимального размера, т. е. кадров длиной 64 байт (без преамбулы), с полем данных в 46 байт. Если скорости указаны для какого-либо определенного протокола, например, Token Ring или FDDI, то они также соответствуют кадрам минимальной длины этого протокола (например, 29 байт для протокола FDDI). Использование в качестве основного показателя скорости работы коммутатора кадров минимальной длины объясняется тем, что такие кадры всегда создают для коммутатора наиболее тяжелый режим работы по сравнению с кадрами другого формата при равной пропускной способности переносимых пользовательских данных. Поэтому при проведении тестирования коммутатора режим передачи кадров минимальной длины используется как наиболее сложный тест, который должен проверить работоспособность коммутатора.

Пропускная способность коммутатора измеряется количеством переданных в единицу времени через его порты пользовательских данных. Так как коммутатор работает на канальном уровне, то для него пользовательскими данными являются те данные, которые переносятся в поле данных кадров протоколов канального уровня – Ethernet, Token Ring, FDDI и т. п. Максимальное значение пропускной способности коммутатора всегда достигается на кадрах максимальной длины, так как при этом и доля накладных расходов на служебную информацию кадра гораздо ниже, чем для кадров минимальной длины, и время выполнения коммутатором операций по обработке кадра, приходящееся на один байт пользовательской информации, существенно меньше.

Зависимость пропускной способности коммутатора от размера передаваемых кадров хорошо иллюстрирует пример протокола Ethernet, для которого при передаче кадров минимальной длины достигается скорость передачи в 14880 кадров в 1с и пропускная способность 5,48 Мбит/с, а при передаче кадров максимальной длины – скорость передачи в 812 кадров в 1с и пропускная способность 9,74 Мбит/с. Пропускная способность падает почти в два раза при переходе на кадры минимальной длины, и это еще без учета потерь времени на обработку кадров коммутатором.

Задержка передачи кадра измеряется как время, прошедшее с момента прихода первого байта кадра на входной порт коммутатора до момента появления этого байта на выходном порту коммутатора. Задержка складывается из времени, затрачиваемого на буферизацию байта кадра, а также времени, затрачиваемого на обработку кадра коммутатором: просмотр адресной таблицы, принятие решения о фильтрации или продвижении и получения 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 10 до 40 мкс, а при полной буферизации кадров – от 50 до 200 мкс (для кадров минимальной длины).

Коммутатор – это многопортовое устройство, поэтому для него принято все приведенные выше характеристики (кроме задержки передачи кадра) давать в двух вариантах. Первый вариант – суммарная производительность коммутатора при одновременной передаче трафика по всем его портам, второй вариант – производительность, приведенная в расчете на один порт.

Так как при одновременной передаче трафика несколькими портами существует огромное количество вариантов трафика, отличающегося размерами кадров в потоке, распределением средней интенсивности потоков кадров между портами назначения, коэффициентами вариации интенсивности потоков кадров и т. д., и т. п., то при сравнении коммутаторов по производительности необходимо принимать во внимание, для какого варианта трафика получены публикуемые данные производительности. К сожалению, для коммутаторов (как, впрочем, и для маршрутизаторов) не существует общепринятых тестовых образцов трафика, которые можно было бы применять для сравнения ха-

рактеристик производительности, как это делается для вычислительных систем.

Максимальная емкость адресной таблицы определяет наибольшее количество MAC-адресов, с которыми может одновременно оперировать коммутатор. Так как коммутаторы чаще всего используют для выполнения операций каждого порта выделенный процессорный блок со своей памятью для хранения экземпляра адресной таблицы, то размер адресной таблицы для коммутаторов обычно приводится в расчете на один порт. Каждый порт хранит только те наборы адресов, которыми он пользуется в последнее время.

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.

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 пакете, то процессор должен вытеснить из таблицы какой-либо старый адрес и поместить на его место новый. Эта операция сама по себе отнимет у процессора часть времени, но главные потери производительности будут наблюдаться при поступлении кадра с адресом назначения, который пришлось удалить из адресной таблицы. Так как адрес назначения кадра неизвестен, то коммутатор должен передать этот кадр на все остальные порты. Эта операция создает лишнюю работу для многих процессоров портов, кроме того, копии этого кадра попадают и в те сегменты сети, где они совсем необязательны.

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. В маршрутизаторах такой прием применяется давно, позволяя сократить размеры адресных таблиц в сетях, организованных по иерархическому принципу. Передача кадра на магистральный порт проводится в расчете на то, что этот порт подключен к вышестоящему коммутатору, имеющему достаточную емкость адресной таблицы и знающему, куда нужно передать любой кадр.

Хотя метод магистрального порта как правило работает эффективно, но существуют ситуации, когда кадры теряют. Например, коммутатор нижнего уровня удалил из своей адресной таблицы адрес MAC_N, который подключен к его порту A, для того, чтобы освободить место для нового адреса MAC_M. При поступлении кадра с адресом назначения MAC_N, коммутатор передает его на магистральный порт B, через который кадр попадает в коммутатор верхнего уровня. Этот коммутатор «видит» по своей адресной таблице, что адрес MAC_N

принадлежит его порту С, через который он и поступил в коммутатор. Поэтому кадр далее не обрабатывается и просто отфильтровывается, а, следовательно, не доходит до адресата.

Внутренняя буферная память коммутатора нужна для временного хранения кадров данных в случаях, когда их невозможно немедленно передать на выходной порт. Буфер предназначен для сглаживания кратковременных пульсаций трафика. Ведь даже если трафик хорошо сбалансирован и производительность процессоров портов, а также других обрабатывающих элементов коммутатора достаточна для передачи средних значений трафика, то это не гарантирует, что их производительности хватит при очень больших нагрузках. Например, трафик может в течение нескольких десятков миллисекунд поступать одновременно на все входы коммутатора, не давая ему возможности передавать принимаемые кадры на выходные порты.

Для предотвращения потерь кадров при кратковременном многократном превышении среднего значения интенсивности трафика (а для локальных сетей часто встречаются значения коэффициента пульсации трафика в диапазоне от 50 до 100) единственным средством служит буфер большого объема. Как и в случае адресных таблиц, каждый процессорный модуль порта обычно имеет свою буферную память для хранения кадров. Чем больше объем этой памяти, тем менее вероятны потери кадров при перегрузках, хотя при несбалансированности средних значений трафика буфер все равно рано или поздно переполниться.

Обычно коммутаторы, предназначенные для работы в ответственных частях сети, имеют буферную память в несколько десятков или сотен килобайт на порт. Хорошо, когда эту буферную память коммутатор может перераспределять между несколькими портами, так как одновременные перегрузки по нескольким портам маловероятны. Дополнительным средством защиты может служить общий для всех портов буфер в модуле управления коммутатором. Такой буфер обычно имеет объем в несколько мегабайт.

Маршрутизаторы

Согласно определению крупнейшего производителя маршрутизаторов компании Cisco маршрутизатор – это устройство третьего уровня, использующее одну и более метрик для определения оптимального пути передачи сетевого трафика на основе информации сетевого уровня. По существу маршрутизатор представляет собой компьютер с необходимым программным обеспечением и устройствами ввода/вывода. В простейшем случае маршрутизатор имеет два сетевых интерфейса.

Маршрутизатор обеспечивает маршрутизацию, т. е. доставку данных адресату, которую можно разбить на три. Во-первых, сбор информации о других маршрутизаторах и хостах в сети. Для этого маршрутизатор в целях определения маршрута использует тот или иной протокол маршрутизации. Во-вторых, он сохраняет полученную информацию о маршрутах в таблицах маршрутизации. В-третьих, маршрутизатор выбирает наилучший маршрут для каждого

конкретного пакета, при этом он передает пакет со входного интерфейса на соответствующий выходной интерфейс. Данные функции он выполняет с помощью протоколов маршрутизации, в основе которых лежат алгоритмы маршрутизации.

Алгоритм маршрутизации – это часть программного обеспечения маршрутизатора, отвечающая за выбор выходной линии, на которую поступивший пакет должен быть передан. Алгоритмы маршрутизации можно разделить на две большие группы: неадаптивные (статические) и адаптивные (динамические). При использовании статических алгоритмов выбор маршрутов осуществляют заранее и прописывают их вручную в таблицу маршрутизации, где хранится информация о том, на какой интерфейс отправить пакет с соответствующей адресной информацией. При использовании динамических алгоритмов таблица маршрутизации меняется автоматически при изменении топологии сети или трафика в ней.

Динамические алгоритмы отличаются по способу получения информации (например, от соседних маршрутизаторов, от всех маршрутизаторов в сети и т. д.), моменту изменения маршрутов (через регулярные интервалы, при изменении топологии и т. п.) и используемой метрике (расстояние, число транзитных узлов и т. д.). Наиболее популярными алгоритмами маршрутизации являются два алгоритма: вектора расстояния и состояния канала.

При использовании алгоритма вектора расстояния каждый маршрутизатор ведет таблицу, т. е. вектор, с указанием кратчайшего расстояния и выходной линии для каждого адресата. В качестве метрики может использоваться также число транзитных узлов, время задержки, совокупная длина очередей и прочее. Таблица содержит информацию обо всех маршрутизаторах в сети. Периодически каждый маршрутизатор рассылает соседям свою таблицу. Одним из основных недостатков этого алгоритма является медленное распространение информации о недоступности той или иной линии или выходе того или иного маршрутизатора из строя. Данный алгоритм используется в таких протоколах, как RIP, IGRP и др.

В случае алгоритма состояния канала маршрутизатор собирает информацию о своих непосредственных соседях, измеряя задержку (пропускную способность). Вместо таблиц маршрутизации он осуществляет широковещательную рассылку информации только о своих непосредственных соседях, причем рассылка инициируется только при изменении информации. При получении изменений маршрутизатор определяет заново кратчайший путь до всех адресатов с помощью алгоритма Э. Дейкстры. Алгоритм состояния канала лежит в основе таких протоколов маршрутизации, как OSPF и IS-IS.

В качества примера таблицы маршрутизации рассмотрим определение таблицы на маршрутизаторе компании Morning Star с тремя интерфейсами: одним интерфейсом Ethernet, последовательным портом, подключенным к внешнему модему, и интерфейсом глобальной сети Frame relay (табл. 3.12). Модем используется для организации связи с сервером в главном офисе по PPP, IP-ад-

рес которого – 137.175.2.7. Адрес интерфейса глобальной сети – 131.187.2.2, а адрес маршрутизатора оператора Internet 131.187.2.3. Для локальной сети оператор выделил блок адресов класса С в диапазоне от 199.18.210.1 до 199.18.210.254. Интерфейсы маршрутизатора поименуем следующим образом: ed0 – интерфейс Ethernet, du0 – последовательный интерфейс, tt0 – интерфейс глобальной сети. Все пакеты, посылаемые в локальную сеть главного офиса, направляются маршрутизатором на последовательный порт. Все пакеты, предназначенные для нашей локальной сети, направляются на интерфейс Ethernet с адресом 199.18.210.1. Адрес 127.0.0.1 является так называемым петлевым адресом, и он используется маршрутизатором для обращения к самому себе. Все остальные пакеты направляются на интерфейс глобальной сети. Флаг U (Up) означает, что соединение активно, а флаг G (Gateway) означает, что шлюз (так изначально назывались маршрутизаторы) действительно является шлюзом в другую сеть, в то время как флаг H (Host) означает, что маршрутизатор подключен к конечному адресату.

Таблица 3.12. **Пример таблицы маршрутизации**

Получатель	Шлюз	Флаг	Интерфейс
default	137.187.2.3	UG	tt0
127.0.0.1	127.0.0.1	UH	lo0
199.18.210.0	199.18.210.0	UG	ed0
137.175.2.7	199.18.210.1	UH	du0
137.187.2.3	137.187.2.2	UH	tt0
137.175.2	137.175.2.7	UG	du0

Поступающие через входной интерфейс пакеты маршрутизатор помещает в очередь (буфер). Для организации очереди маршрутизаторы как правило используют алгоритм «честной очереди» и его модификацию – алгоритм «честной взвешенной очереди» (Weighted Fair Queue). Суть данного алгоритма состоит в том, что маршрутизаторы имеют несколько очередей для каждой выходной линии, по одной для каждого отправителя. Когда линия освобождается, маршрутизатор берет пакет из следующей по кругу очереди. Модифицированный же алгоритм позволяет давать приоритет тем или иным очередям.

Маршрутизаторы могут работать с пакетами нескольких протоколов сетевого уровня. Многопротокольные маршрутизаторы концептуально напоминают мосты, но с той существенной разницей, что они работают на сетевом уровне. Как и любой маршрутизатор, они берут пакет с одной линии и передают его на другую, но при этом линии принадлежат к разным сетям и используют разные протоколы (например, IP и IPX). Кроме того, сетевые устройства типа моста/маршрутизатора (brouter или bridge/router) работают в нормальном режиме как многопротокольные маршрутизаторы, а при получении пакета с неизвестным сетевым протоколом обрабатывают его как мост. Устройства со сходным названием «маршрутизирующий мост» (routing bridge) принадлежат к устройствам второго уровня и упоминаются здесь лишь из-за причастия routing. Они рабо-

тают как мосты, но при этом поддерживают некоторые функции третьего уровня для оптимизации передачи данных.

Маршрутизаторы с интеграцией услуг гарантируют приоритетному трафику, в частности трафику реального времени, своевременную доставку. Они поддерживают протокол RSVP для резервирования пропускной способности и буфера в очереди. Коммутаторы третьего уровня по сути также являются маршрутизаторами, причем пакетные коммутаторы (Packet-by-Packet Switch), на самом деле являются обычными, только быстрыми маршрутизаторами.

Любой маршрутизатор имеет несколько интерфейсов к локальным сетям (LAN) и несколько интерфейсов к сетям глобальной топологии (WAN). В качестве интерфейсов LAN сегодня выступают, как правило, Ethernet или Fast Ethernet. Выбор возможных интерфейсов WAN шире. Для подключения к коммуникационному оборудованию, например модемам или терминальным адаптерам ISDN, чаще всего используют последовательные интерфейсы. Они могут быть синхронными или асинхронными. Самый популярный из них – низкоскоростной асинхронный интерфейс RS-232 (COM-порт).

Маршрутизатор может подключаться к различным сетям и напрямую. Некоторые маршрутизаторы имеют интерфейсы ISDN – так называемые BRI (144 Кбит/с) или PRI (2 Мбит/с). Есть маршрутизаторы со встроенными модемами, подключающиеся к телефонной линии.

Маршрутизаторы имеют много разных параметров производительности важнейшим из которых является пропускная способность. Она измеряется в пакетах в секунду (пак/с – pps), как правило, для самых маленьких пакетов (64 байт). Некоторые маршрутизаторы заявляют огромную производительность – десятки миллионов пакетов в секунду, но для того, чтобы загрузить на 100% один асинхронный последовательный канал 115,2 кбит/с, достаточно производительности 235 пак/с. Для того чтобы маршрутизатор не вносил задержек при маршрутизации между двумя локальными сетями Ethernet, достаточно 14880 пак/с, для Fast Ethernet – 148800 пак/с.

Маршрутизаторы часто путают с мостами. Это объясняется тем, что многие устройства сочетают в себе функции и мостов, и маршрутизаторов. «Чистый» мост анализирует заголовки кадра канального уровня и не просматривает (а тем более не модифицирует) пакеты сетевого уровня внутри пакетов. Мост не знает и не должен знать, какие пакеты (IP, IPX или CLNP) содержат в поле полезной нагрузки кадр, передаваемый из локальной сети 802.x в 802.y. Маршрутизатор, наоборот, знает очень хорошо, с какими пакетами он работает – с IP, IPX, CLNP – или со всеми ними сразу (в случае многопротокольных маршрутизаторов). Он анализирует заголовки этих пакетов и принимает решение в соответствии с содержащейся там адресной информацией. С другой стороны, когда «чистый» маршрутизатор передает пакет на канальный уровень, он не знает и не должен знать о том, в какой кадр данный пакет будет помещен в Ethernet, Token Ring или какой-либо другой. Путаница происходит по двум причинам. Во-первых, по части функциональности мосты и маршрутизаторы весьма

напоминают друг друга. Они принимают протокольные блоки данных (PDU – Protocol Data Unit), анализируют определенные поля заголовка и принимают решение о том, куда дальше передать пакет в соответствии с содержащейся в заголовке информацией и внутренними таблицами. Во-вторых, названия коммерческим продуктам даются зачастую весьма условные, а, кроме того, многие из них сочетают в себе функции и тех и других устройств.

В последние несколько лет сама необходимость в маршрутизаторах начала подвергаться сомнению, главным образом в связи с появлением и распространением коммутаторов (по сути многопортовых мостов). В чем же причины того, что пользователи начали устанавливать в своих сетях коммутаторы там, где они раньше использовали маршрутизаторы? Вот некоторые из них: маршрутизаторы в расчете на порт стоят гораздо дороже коммутаторов; как правило, в сравнении с коммутаторами они имеют гораздо меньшую совокупную пропускную способность (пакетов в секунду), функции коммутации второго уровня гораздо проще реализовать аппаратным образом, чем программным, и т. д. Однако основным недостатком маршрутизаторов по сравнению с коммутаторами является то, что последние требуют гораздо меньших усилий по администрированию. Сетевым администраторам приходится задавать целое множество конфигурационных параметров для каждого маршрутизатора в сети, таких как адреса и маски подсети, статические маршруты и т. д. Еще хуже то, что параметры каждого маршрутизатора должны быть согласованы с параметрами других маршрутизаторов в сети.

С другой стороны, маршрутизаторы выполняют многие функции, с которыми коммутаторы справиться, как правило, не в состоянии, так как они функционируют на другом уровне. Например, маршрутизаторы позволяют решить такую типичную проблему при связи сетей с помощью мостов, как штормы широковещательных пакетов. Кроме того, маршрутизаторы используются зачастую в качестве брандмауэров (защитных экранов) между корпоративной сетью и Internet. При этом они действуют как фильтры пакетов, просматривая адресную информацию заголовка пакета и сопоставляя ее со списком управления доступом. Далее, маршрутизаторы могут применяться для фильтрации трафика по каналам глобальной сети, передавая через нее только избранный трафик, что, в частности, позволяет снизить плату за использование этих каналов. Во многом благодаря перечисленным функциям маршрутизаторам, а не мостам, было в свое время отдано предпочтение. Поэтому использование коммутаторов (без маршрутизации) должно быть хорошо продумано, иначе могут возникнуть дополнительные проблемы.

Сравнение сетевых устройств

Повторители Ethernet, в контексте сетей 10Base-T часто называемые концентраторами или хабами, работают в соответствии со стандартом IEEE 802.3. Повторитель передает полученные пакеты во все свои порты независимо от адресата.

Хотя все устройства, подключенные к повторителю Ethernet (включая другие повторители) «видят» весь сетевой трафик, получить пакет должен только тот узел, которому он адресован, а все остальные узлы должны игнорировать этот пакет. Некоторые сетевые устройства (например, анализаторы протоколов) работают на основе того, что сетевая среда (типа Ethernet) является общедоступной и анализируют весь сетевой трафик.

С точки зрения производительности повторители просто передают пакеты с использованием всей полосы канала. Задержка, вносимая повторителем весьма мала (в соответствии с IEEE 802.3 – менее 3 мс). Сети, содержащие повторители, имеют полосу 10 Мбит/с подобно сегменту на основе коаксиального кабеля и прозрачны для большинства сетевых протоколов, таких как TCP/IP и IPX.

Мосты функционируют в соответствии со стандартом IEEE 802.1d. Подобно коммутаторам Ethernet мосты не зависят от протокола и передают пакеты порту, к которому подключен адресат. Однако, в отличие от большинства коммутаторов Ethernet, мосты не передают фрагменты пакетов при возникновении коллизий и пакеты с ошибками, поскольку все пакеты буферизуются перед их пересылкой в порт адресата. Буферизация пакетов (store-and-forward) приводит к возникновению задержки по сравнению с коммутацией на лету. Мосты могут обеспечивать производительность, равную пропускной способности среды, однако внутренняя блокировка несколько снижает скорость их работы.

Работа *маршрутизаторов* зависит от сетевых протоколов и определяется связанной с протоколом информацией, передаваемой в пакете. Подобно мостам, маршрутизаторы не передают адресату фрагменты пакетов при возникновении коллизий. Маршрутизаторы сохраняют пакет целиком в своей памяти прежде, чем передать его адресату, следовательно, при их использовании пакеты передаются с задержкой. Маршрутизаторы могут обеспечивать полосу, равную пропускной способности канала, однако для них характерно наличие внутренней блокировки. В отличие от повторителей, мостов и коммутаторов маршрутизаторы изменяют все передаваемые пакеты.

4. ТЕХНОЛОГИИ ГЛОБАЛЬНЫХ СЕТЕЙ

Описаны основные технологии глобальных сетей X.25, Frame relay, ISDN, ATM и сетей мобильной связи. Приведены сведения об основах каждой технологии , форматах кадров и протоколах. Определены области применения и возможности по предоставлению услуг пользователям. Отмечены аспекты безопасности в сетях сотовой мобильной связи. Рассмотрена структура и функциональное назначение устройств сети спутниковой мобильной связи на примере системы Глобалстар.

4.1. Технология X.25

Технология X.25 (официально называемая CCITT Recommendation X.25 – «Рекомендация X.25» CCITT) – международный стандарт передачи пакетов по общественным сетям. Он поддерживает линии передачи данных со средней или высокой скоростью передачи для постоянного или периодического использования. Стандарт X.25, как правило, используют для организации международных сетей.

Для связи локальной сети с сетью X.25 используется мост или маршрутизатор. Доступ к сети осуществляется через арендуемую линию или линию с вызовом по номеру. В выделенных линиях обычно используют синхронную связь, что увеличивает пропускную способность. Скорость передачи составляет 19,2...64 кбит/с. Линии с вызовом по номеру используют асинхронные методы с применением модемов, которые имеют собственные средства коррекции ошибок. Скорость передачи зависит от скорости модема.

Сети с коммутацией пакетов X.25 не обеспечивают качественную передачу критичного к задержкам трафика, так как в них отсутствуют механизмы обеспечения приоритетов каких-либо видов данных. Дело в том, что технология X.25 предназначена для организации надежной передачи данных в условиях разветвленных территориально-распределенных сетей на базе низко- и среднескоростных каналов невысокого качества. При этом обеспечивается достоверная и упорядоченная (за счет повторной передачи искаженных кадров) передача данных между каждой парой соседних узлов сети по всему мар-

шруту следования пакета. В сетях с каналами низкого качества возникают нерегламентированные непостоянные по величине задержки передаваемых данных.

Основы технологии

Технология X.25 определяет характеристики телефонной сети для передачи данных. Чтобы начать связь, один компьютер обращается к другому с запросом о сеансе связи. Вызванный компьютер может принять или отклонить связь. Если вызов принят, то обе системы могут начать передачу информации с полным дублированием. Любая сторона может в любой момент прекратить связь. Спецификация X.25 определяет двухточечное взаимодействие между терминальным оборудованием (DTE) и оборудованием завершения действия информационной цепи (DCE). Устройства DTE (терминалы и главные вычислительные машины – хосты) подключают к устройствам DCE (модемы, коммутаторы пакетов и другие порты в сети PDN), которые соединены с «коммутаторами переключения пакетов» (PSE – packet switching exchange или просто switches) и другими DCE внутри сети с коммутацией пакетов PSN и, наконец, с другим устройством DTE. Взаимоотношения между объектами сети X.25 показаны на рис.4.1.

DTE может быть терминалом, который не полностью реализует все функциональные возможности X.25. Такие DTE подключают к DCE через трансляционное устройство, называемое пакетный ассемблер/дизассемблер (PAD – packet assembler/disassembler). Их используют для доступа в сеть абонентов в асинхронном режиме обмена информацией, т. е., например, через последовательный порт компьютера (непосредственно или с применением модемов). PAD обычно имеет несколько асинхронных портов и один синхронный (порт X.25). Он накапливает поступающие через асинхронные порты данные, упаковывает их в пакеты и передает через порт X.25.

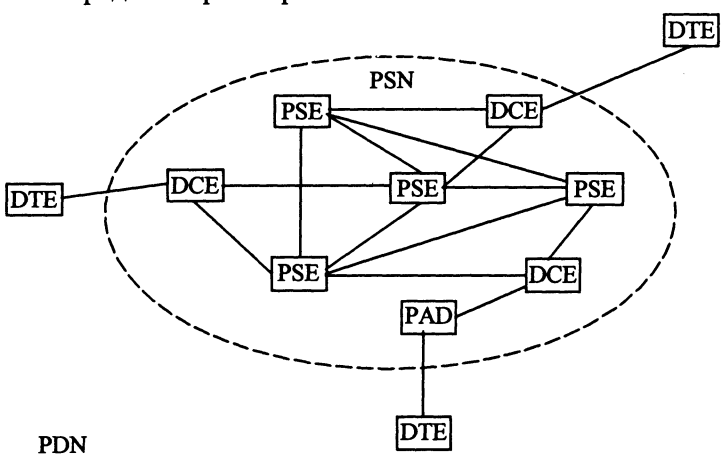


Рис. 4.1. Модель сети X.25

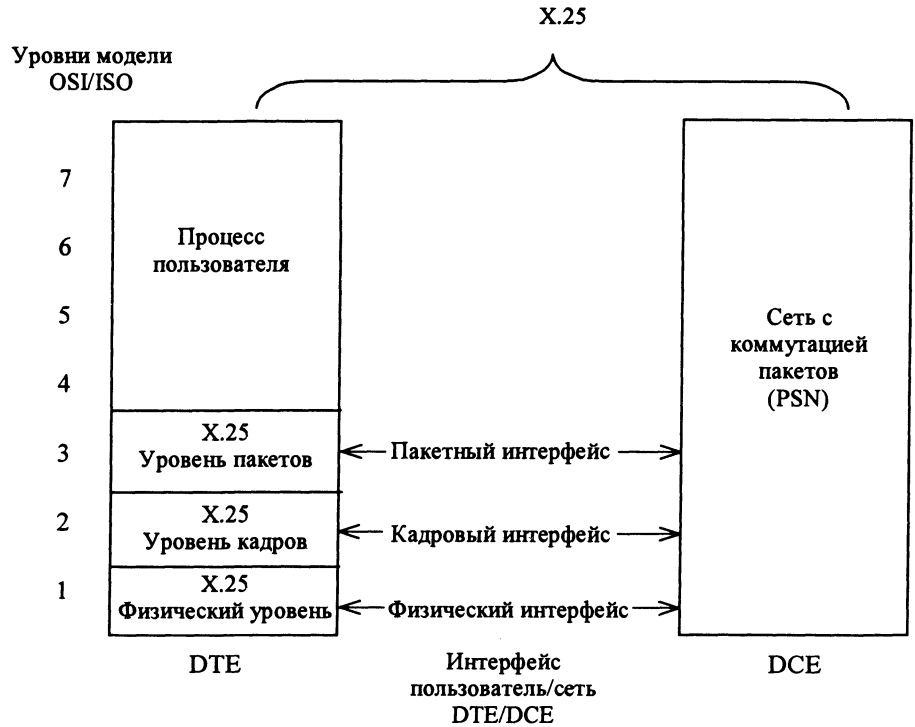


Рис. 4.2. Соответствие между уровнями X.25 и моделью OSI/ISO

Интерфейс терминал/PAD, услуги, предлагаемые PAD, взаимодействие между PAD и главной вычислительной машиной определены рекомендациями CCITT X.28, X3 и X.29 соответственно.

Спецификация X.25 соответствует первым трем уровням эталонной модели OSI. Уровень 3 X.25 описывает форматы пакетов и процедуры обмена пакетами между равноправными объектами этого уровня. Уровень 2 X.25 реализован протоколом Link Access Procedure, Balanced (LAPB), определяющим кадрирование пакетов для звена DTE/DCE. Уровень 1 X.25 определяет электрические и механические процедуры активации и дезактивации физической среды, соединяющей данные DTE и DCE (рис. 4.2). Необходимо отметить, что на Уровни 2 и 3 также ссылаются как на стандарты ISO – ISO 7776 (LAPB) и ISO 8208 (пакетный уровень X.25).

Сквозная передача между устройствами DTE выполняется через двунаправленную связь, называемую виртуальной цепью. Виртуальные цепи позволяют осуществлять связь между различными элементами сети через любое число промежуточных узлов без назначения частей физической среды, что характерно для физических цепей. Виртуальные цепи бывают постоянные, или коммутируемые (временные). Постоянные виртуальные цепи называют PVC; ком-

мутируемые виртуальные цепи – SVC. Уровень 3 X.25 отвечает за сквозную передачу, включающую как PVC, так и SVC-цепи.

После организации виртуальной цепи DTE отсылает пакет на другой конец связи через DCE, используя соответствующую виртуальную цепь. DCE просматривает номер виртуальной цепи для определения маршрута этого пакета через сеть X.25. Протокол Уровня 3 X.25 осуществляет мультиплексную передачу между всеми DTE, которые обслуживает устройство DCE, расположенное в сети со стороны пункта назначения, в результате чего пакет доставляется к DTE пункта назначения.

Формат блока данных

Блок данных X.25 состоит из последовательности полей, показанной на рис. 4.3. Поля X.25 Уровня 3 образуют пакет X.25; они состоят из заголовка и данных пользователя. Поля X.25 Уровня 2 (LAPB) включают в себя поле управления и адреса кадра, встроенный пакет Уровня 2 (поле данных) и проверочную последовательность блока данных (FCS).

Заголовок X.25 Уровня 3 образован из идентификатора универсального формата (GFI – general format identifier), идентификатора логического канала (LCI – logical channel identifier) и идентификатора типа пакета (PTI – Packet Type Identifier). GFI представляет собой 4-битовое поле, которое указывает на универсальный формат заголовка пакета, LCI – 12-битовое поле, идентифицирующее виртуальную цепь. Поле LCI является логически значимым в интерфейсе DTE/DCE. Другими словами, для организации виртуальной цепи PDN соединяет два логических канала, каждый из которых имеет независимый LCI, двумя интерфейсами DTE/DCE. Поле PTI идентифицирует один из 17 типов пакетов X.25.

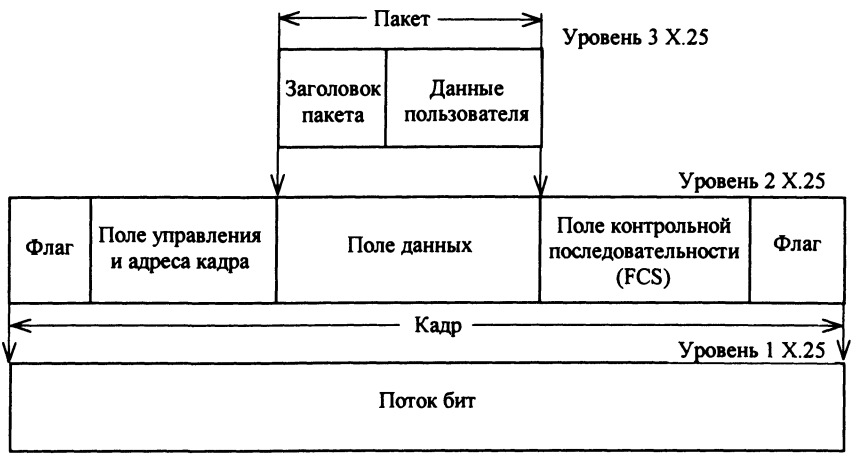


Рис. 4.3. Формат блока данных X.25

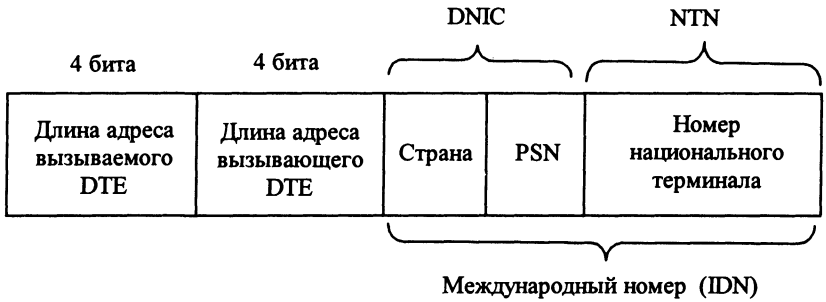


Рис. 4.4. Формат адреса X.121

Поля адресации в пакетах запроса на установление соединения содержат адреса DTE источника и пункта назначения. Их используют для организации виртуальных цепей. Рекомендация X.121 CCITT определяет форматы адресов источника и пункта назначения. Адреса X.121 (называемые также International Data Numbers, или IDN) имеют разную длину, которая может составлять до 14 десятичных знаков. Четвертый байт в пакете запроса на установление соединения определяет длину адресов DTE источника и назначения. Первые четыре цифры IDN называются код идентификации сети (DNIC – Data Network Identification Code). Он поделен на две части: первая часть (3 цифры) определяет страну, где находится PSN, вторая часть – саму PSN. Остальные цифры называются номером национального терминала (NTN – national terminal number); их используют для идентификации определенного DTE в сети PSN. Формат адреса X.121 представлен на рис. 4.4.

В сетях в соответствии с Рекомендацией X.121 используют адреса 3 типов:

1. Полный (международный) сетевой адрес – 0 250 C XXXXYYYYZZ, где (слева направо):

· 0 – признак того, что адрес задан в полном виде;

· 2504 DNIC – код сети, где:

· 2 – код Европы;

· 50 – код страны (СНГ/СССР);

· C – код национальной сети (например, для ИНФОТЕЛ C = 4);

· XXXXYYYYZZ – уникальный код абонента внутри сети (NTN – до 10 цифр).

2. Внутрисетевой адрес – CXXXXXXXXXX, где:

· C – последняя цифра DNIC-а, т. е. уникальный код сети внутри одной страны;

· XXXXXXXXXX – код абонента.

3. Телефонный номер – 9GNNNNNNNNNN, где:

· 9 – признак телефонного номера;

· G – код узла коммутации (города);

· NNNNNNNNNN – телефонный номер внутри города (до 11 цифр).

Поля адресации, образующие адрес X.121, необходимы только при использовании SVC на время установления вызова. После того, как вызов организован, PSN использует поле LCI заголовка пакета данных для назначения конкретной виртуальной цепи отдаленному DTE.

Основные процедуры X.25

X.25 Уровня 3 использует три рабочих процедуры организации виртуальной цепи:

- установление соединения;
- передача данных;
- разъединение вызова.

Выполнение этих процедур зависит от типа виртуальной цепи. Для PVC Уровень 3 X.25 всегда находится в режиме передачи данных, так как цепь организована постоянно. Если необходимо организовать SVC, то реализует все три процедуры.

Процедура передачи данных зависит от принятой пакетов DATA. Протокол X.25 Уровня 3 сегментирует и разбивает сообщения пользователя, если их длина превышает максимальный размер пакета для данной виртуальной цепи. Каждому пакету DATA присваивается номер последовательности, поэтому можно управлять неисправностями и потоком информации через интерфейс DTE/DCE.

Протокол LAPB

Уровень 2 реализован протоколом LAPB, позволяющим обеим сторонам (DTE и DCE) инициировать связь друг с другом. В процессе передачи информации LAPB контролирует, чтобы блоки данных поступали к приемному устройству в правильной последовательности и без ошибок.

Протокол LAPB так же, как и аналогичные протоколы канального уровня, использует три типа форматов блоков данных.

- Информационные блоки данных (Information (I) frame). Эти блоки данных содержат информацию высших уровней и определенную управляющую информацию (необходимую для работы с полным дублированием). Номера последовательности отправки и приема осуществляют управление информационным потоком. Номер последовательности отправки относится к номеру текущего блока данных. Номер последовательности приема фиксирует номер блока данных, который должен быть принят следующим. В диалоге с полным дублированием как отправитель, так и получатель хранят номера последовательности отправки и приема; она используется для обнаружения и устранения ошибок.
- Блоки данных супервизора (Supervisory (S) frames). Такие блоки данных обеспечивают управляющую информацию. У них нет информационного поля. Блоки данных S запрашивают и приостанавливают передачу, сообщают о состоянии канала и подтверждают прием блоков данных типа I.

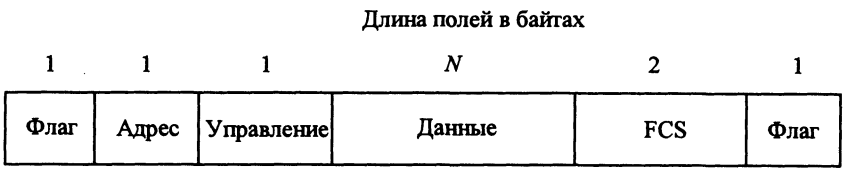


Рис. 4.5. Блок данных LAPB

• Ненумерованные блоки данных (Unnumbered (U) frames). Их применяют для управляющих целей. С их помощью можно инициировать связи, используя стандартную (mod 8) или расширяемую (mod 128) организацию окон, разъединять канал, сообщать об ошибках в протоколе и выполнять другие аналогичные функции. Блок данных LAPB представлен на рис. 4.5.

Поле флаг ограничивает блок данных LAPB. Чтобы предотвратить появление структуры флага в пределах внутренней части блока данных, используют вставку битов. Поле *адрес* указывает, что содержит блок данных – команду или ответный сигнал. Поле *управление* обеспечивает дальнейшую классификацию блоков данных и блоков команд, а также указывает формат блока данных (U, I или S), функции блока данных (например, receiver ready – получатель готов или disconnect – отключение) и номер последовательности передачи/приема.

Поле *данные* содержит данные высших уровней. Его размер и формат меняются в зависимости от типа пакета Уровня 3. Максимальная длина этого поля устанавливается соглашением между администратором PSN и абонентом во время регистрации абонента. Поле FCS обеспечивает целостность передаваемых данных.

Уровень 1 X.25 использует протокол физического уровня X.21 bis, который примерно эквивалентен RS-232-C. Протокол X.21 bis является производным Рекомендаций V24 и V25 CCITT, которые соответственно идентифицируют цепи обмена и характеристики электрических сигналов интерфейса DTE/DCE. Протокол физического уровня X.21 bis обеспечивает двухточечные связи, скорости до 19,2 Кб/с и синхронную передачу с полным дублированием через 4-проводной носитель. Максимальное расстояние между DTE и DCE составляет 15 м.

На сегодняшний день накоплен большой опыт использования сетей X.25, который показывает, что они эффективны для широкого круга задач передачи данных: обмен сообщениями, обращение большого количества пользователей к удаленной базе данных, связь локальных сетей (при ограничении скорости не более 512 кбит/с), объединение удаленных кассовых аппаратов и банкоматов и пр. Все эти случаи, да и другие, не указанные, объединяет то, что трафик в сети не является равномерным во времени. Немаловажным достоинством сетей X.25 является то, что по ним можно передавать данные по каналам телефонной сети общего пользования, как выделенным, так и коммутируемым с максимальной для этих каналов скоростью и достоверностью.

Кроме того, сети X.25 предоставляют возможность связи через обычные асинхронные COM-порты. Таким образом, практически любое приложение, допускающее обращение к удаленным ресурсам через COM-порт, может быть легко интегрировано в сеть X.25. В качестве примеров можно упомянуть терминальный доступ к удаленным хост-компьютерам, например Unix – машинам, взаимодействие друг с другом Unix – компьютеров, электронную почту Lotus cc:Mail, MS Mail и др.

Для объединения локальных сетей в узлах, имеющих подключение к сети X.25, используют методы инкапсуляции (упаковки) пакетов информации из локальной сети в пакеты X.25. При этом часть служебной информации, которая может быть однозначно восстановлена на стороне получателя, не передается. Стандартный механизм инкапсуляции описан в документе RFC 1356. Он позволяет передавать различные протоколы локальных сетей (IP, IPX и др.) одновременно через одно виртуальное соединение и реализован (иногда в более ранней модификации RFC 877, позволяющей передавать только IP) практически во всех современных маршрутизаторах. Применяют также методы передачи по X.25 других коммуникационных протоколов, в частности SNA, используемого в сетях IBM mainframe, а также ряда частных протоколов различных производителей. Таким образом, сети X.25 предлагают универсальный транспортный механизм для передачи информации между практически любыми приложениями. При этом разные типы трафика передаются по одному каналу связи, ничего не «зная» друг о друге. При объединении локальных сетей через X.25 отдельные фрагменты корпоративной сети, даже и использующие одни и те же линии связи, можно изолировать друг от друга, что облегчает решение проблемы безопасности и разграничения доступа, которые возникают в сложных информационных структурах. Во многих случаях отпадает необходимость использования сложных механизмов маршрутизации, так как эту функцию может выполнять сеть X.25.

Эффективным механизмом оптимизации процесса передачи информации через сети X.25 является механизм альтернативной маршрутизации. Возможность задания помимо основного маршрута альтернативных, т. е. резервных, предусмотрена в оборудовании X.25, производимом практически всеми фирмами. Различные образцы оборудования отличаются алгоритмами перехода к альтернативному маршруту, а также их допустимым количеством. Переход к альтернативному маршруту происходит либо в случае полного отказа одного из звеньев основного маршрута, либо динамически в зависимости от загруженности маршрутов, и решение принимается на основании многопараметрической формулы (например оборудование фирмы Motorola ISG). За счет альтернативной маршрутизации можно значительно увеличить надежность работы сети, а это значит, что между любыми двумя точками подключения пользователя к сети должно быть, по крайней мере, два различных маршрута.

Сегодня в мире насчитывают десятки глобальных сетей X.25 общего пользования, узлы которых расположены практически во всех крупных деловых, про-

мышленных и административных центрах. В России услуги X.25 предлагает ряд компаний, таких, как Sovam Teleport, Infotel, Роснет и др. В сетях X.25 кроме объединения удаленных узлов предусмотрены средства доступа конечных пользователей. Для того чтобы подключиться к любому ресурсу сети X.25, пользователю достаточно иметь компьютер с последовательным асинхронным портом и модем. С авторизацией доступа в различных географически удаленных узлах проблем не возникает, во-первых, потому, что сети X.25 достаточно централизованы и, заключив договор, скажем, с компанией SprintNet или ее партнером, можно пользоваться услугами любого из узлов SprintNet – а это тысячи городов по всему миру, в том числе более сотни на территории бывшего СССР. Во-вторых, существует протокол взаимодействия между разными сетями (описанный в рекомендации X.75 МККТТ), учитывающий, в том числе, и вопросы оплаты. Таким образом, пользователь, подключенный к сети X.25, имеет возможность получить доступ к ресурсам сети как с узлов своего поставщика, так и через узлы других сетей.

С точки зрения безопасности передачи информации, сети X.25 имеют ряд достоинств. Во-первых, благодаря самой структуре сети X.25 стоимость перехвата информации оказывается достаточно высокой, что само по себе является неплохой защитой. С помощью самой сети можно эффективно решить проблему несанкционированного доступа. В случае же, если необходима полная конфиденциальность, когда неприемлем даже небольшой риск перехвата информации, необходимо использовать средства шифрования, в том числе и в реальном времени. В настоящее время для сетей X.25 разработаны средства шифрования, позволяющие работать на достаточно высоких скоростях (до 64 кбит/с). Такое оборудование производят компании Racal, Cylink, Siemens. Есть и российские разработки, созданные под эгидой ФАПСИ.

В настоящее время, правда, принято считать, что сети X.25 медленны, дороги и вообще устарели. Практически не существует сетей X.25, использующих скорости, превышающие 128 кбит/с. Связано это с тем, в частности, что протокол X.25 включает в себя мощные средства коррекции ошибок, обеспечивая передачу данных без искажений даже на линиях плохого качества. Следует особо отметить тот факт, что в России, к сожалению, каналов хорошего качества нет практически нигде. Понятно, что за надежность связи приходится платить, как правило, именно быстродействием оборудования сети и сравнительно большими, хотя и предсказуемыми, задержками распространения информации. Кроме того, протокол X.25 достаточно универсален и позволяет передавать практически любые типы данных. Для сетей X.25 «естественным» является работа приложений, использующих стек протоколов OSI, а именно: системы, работающие в соответствии со стандартом X.400 (электронная почта), FTAM (обмен файлами) и др. Доступны средства, позволяющие реализовать на базе протоколов OSI взаимодействие Unix-систем.

Недостатками технологии X.25 является наличие ряда принципиальных ограничений по скорости. Одно из них связано с весьма развитыми возможнос-

тями коррекции ошибок. Эти средства вызывают задержки передачи информации и требуют от аппаратуры X.25 большой вычислительной мощности и производительности. Несмотря на то, что существует оборудование, имеющее двухмегабитные порты, реально обеспечиваемая ими скорость не превышает 250...300 кбит/с на порт. Для современных скоростных линий связи средства коррекции X.25 избыточны, и при их использовании мощности оборудования зачастую работают вхолостую.

Второй недостаток, заставляющий рассматривать сети X.25 как медленные, заключается в особенностях инкапсуляции протоколов локальных сетей (главным образом IP и IPX). При прочих равных условиях связь локальных сетей по X.25 оказывается на 15...40 % (в зависимости от параметров сети) медленнее, чем при использовании HDLC по выделенной линии. Причем, чем хуже линия связи, тем выше потери производительности. Это также связано с очевидной избыточностью: протоколы LAN имеют собственные средства коррекции и восстановления (TCP, SPX), однако при использовании X.25 приходится делать это еще раз, теряя скорость.

Именно на основании этих недостатков сети X.25 считают медленными и устаревшими. Тем не менее, на линиях невысокого качества сети X.25 вполне эффективны и дают значительный выигрыш по цене и возможностям по сравнению с выделенными линиями, хотя по ним невозможно передавать голос и видео. С другой стороны, даже рассчитывая на быстрое улучшение качества связи вложения в аппаратуру X.25 не пропадут, так как современное оборудование включает возможность перехода к технологии Frame relay.

4.2. Технология ISDN

Согласно определению Международного Союза Связи (ITU – International Telecommunications Union), головной организации по разработке телекоммуникационных стандартов, ISDN представляет собой набор стандартных интерфейсов для цифровой сети связи. По своей сути ISDN – это цифровой вариант аналоговых телефонных линий с коммутацией цифровых потоков или, иначе, сеть из цифровых телефонных станций, соединенных друг с другом цифровыми каналами.

Рассмотрим более подробно интерфейсы ISDN и средства подключения удаленных ПК и ЛВС к ISDN-сети. Одним из основных элементов любой коммуникационной системы являются линии связи. В ISDN использовано несколько принципиально различных типов соединительных линий, или интерфейсов.

Интерфейсы ISDN

Рост нагрузки на аналоговые линии связи, обусловленный интенсификацией информационных потоков и расширением круга задач, возлагаемых на телефонную связь, привел к необходимости выбора: либо повысить кабельную емкость путем увеличения количества линий связи (что приведет к значительному удорожанию телефонных услуг), либо искать принципиально новые решения. В результате появились цифровые линии связи (DTI – Digital Trunk Interface),

позволившие увеличить количество каналов при сохранении или даже уменьшении числа соединительных проводов. Первые ISDN-станции (70-е годы XX в.) разрабатывались с учетом возможности работы с аналоговыми линиями связи и DTI. Дальнейшее развитие цифровых принципов связи привело к увеличению численности ISDN-станций, что, в свою очередь, вызвало необходимость создания специфического ISDN-интерфейса, обеспечивающего связь между ISDN-станциями. При этом физическая совместимость нового ISDN-интерфейса с DTI позволяет абонентам ISDN-станций наряду с ISDN-терминалами использовать аналоговые телефонные аппараты, модемы и факсы. Подключение внешних устройств ISDN и связь между ISDN станциями показана на рис. 4.6.

В ISDN-сетях используются два специфических типа интерфейсов: интерфейс базового уровня BRI (Basic Rate Interface), регламентирующий соединение ISDN-станции с абонентом, и интерфейс первичного уровня PRI (Primary Rate Interface), обеспечивающий связь между ISDN-станциями. Логически BRI представляет собой особым образом структурированный цифровой поток, разделенный на три канала: два информационных канала типа B с пропускной способностью 64 кбит/с каждый и один служебный канал типа D с пропускной способностью 16 кбит/с. Именно поэтому BRI имеет еще одно наименование – 2B + D-интефейс. При использовании BRI в качестве связующего звена между ISDN-станцией и цифровым телефонным аппаратом по B-каналам передают оцифрованные речевые сигналы, при организации же удаленного доступа к ПК и ЛВС или выхода в Internet B-каналы используют для обмена данными. При этом по одной линии BRI можно передавать два независимых потока сообщений – по числу B-каналов.

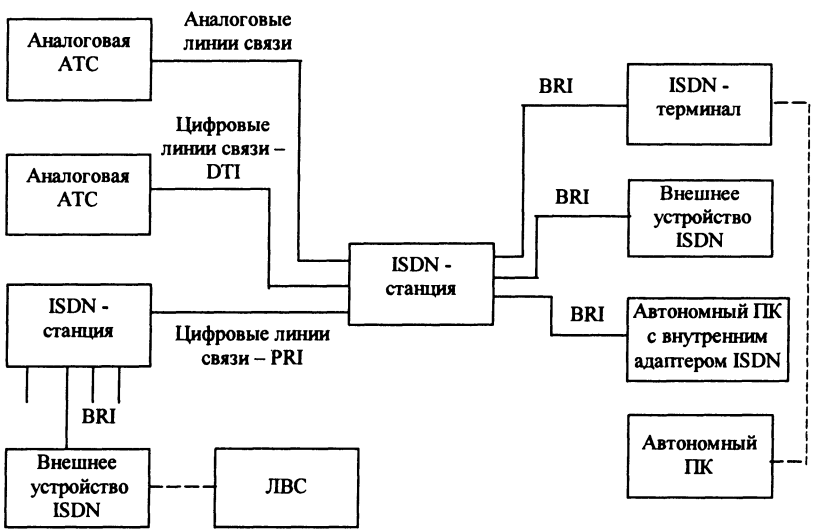


Рис. 4.6. Пример подключения внешних устройств в ISDN

D-канал, как уже говорилось выше, выполняет служебные функции: передача служебной информации (сигналы вызова, маршрут звонка, номера вызываемого и вызывающего абонентов и т. д.), одновременное обслуживание нескольких B-каналов, контроль занятости B-каналов, присвоение каждому абоненту определенного имени (при включении данного абонента в базу данных на ISDN-станции), вывод номера и имени звонящего абонента на экран дисплея ISDN-терминала и многое другое.

Протокол обмена сигналами D-канала включает Уровни 1 – 3 эталонной модели OSI. BRI обеспечивает также управление разметкой и другие операции, не связанные с передачей данных, при этом общая скорость передачи битов доходит до 192 кбит/с. Спецификацией физического уровня BRI является рекомендация CCITT 1.430.

ISDN-станции, в которые «стекаются» BRI-интерфейсы, соединены между собой широкополосными магистралями, поддерживающими интерфейс первичного уровня PRI. Логически PRI построен по тому же принципу, что и BRI-интерфейс: определенное количество B-каналов и один D-канал. Иными словами, PRI можно представить в виде формулы nB+D (23B+D в США и Японии, где действует стандарт T1, и 30B+D в Европе, где действует стандарт E1). При этом следует помнить, что D-каналы в PRI и BRI отличаются пропускной способностью: если в BRI быстродействие D-канала равно 16 кбит/с, то в PRI – 64 кбит/с. Услуги PRI (интерфейса первичной скорости) ISDN предлагают 23 B-канала и 1 D-канал в Северной Америке и Японии, обеспечивающие общую скорость передачи битов 1,544 Мбит/с (D-канал PRI работает на скорости 64 кбит/с). PRI ISDN в Европе, Австралии и других частях света обеспечивает 30 B-каналов, 1 64 кбит/с D-канал и общую скорость интерфейса 2,048 Мбит/с. Спецификацией физического уровня PRI является CCITT 1.431.

Технологией ISDN определены два типа терминалов. Специализированные терминалы ISDN называют «терминальным оборудованием типа 1» TE1 (terminal equipment type 1). Терминалы, появившиеся раньше стандартов ISDN, например DTE, называют «терминальным оборудованием типа 2» TE2 (terminal equipment type 2). Терминалы TE1 подключают к сети ISDN через цифровую линию связи из четырех скрученных пар проводов, а TE2 – через терминальный адаптер. Терминальный адаптер (TA) ISDN может быть автономным устройством, либо платой внутри TE2. Если TE2 реализован как автономное устройств, то его подключают к TA через стандартный интерфейс физического уровня (например, RS-232, V.24/V.28 или V.35).

Следующей точкой соединения в сети ISDN является устройство завершения работы сети NT1 или NT2, которое подключают четырехпроводной абонентский монтаж к традиционному контуру двухпроводной сети. В Северной Америке NT1 является устройством «оборудования посылок заказчика» CPE (customer premises equipment). В большинстве других частей света NT1 является частью сети, обеспечиваемой коммерческими сетями связи. NT2 явля-

ется более сложным устройством, которое обычно применяют в «частных цифровых телефонных станциях с выходом в общую сеть» (PBX). Оно выполняет функции протоколов Уровней 2 и 3 и услуги по концентрации данных. Существует устройство NT1/2, сочетающее функции NT1 и NT2.

В ISDN задано определенное число контрольных точек. Они определяют логические интерфейсы между функциональными группировками, такими, как TA и NT1. Контрольными точками ISDN являются точки R (контрольная точка между неспециализированным оборудованием ISDN и TA), S (контрольная точка между терминалами пользователя и NT2), T (контрольная точка между устройствами NT1 и NT2) и U (контрольная точка между устройствами NT1 и оборудованием завершения работы линии в коммерческих сетях связи). Контрольную точку U используют только в Северной Америке, где функция NT1 не обеспечивается коммерческими сетями связи.

Физически BRI может быть реализовано в виде U- или S/T-интерфейса. U-интерфейс предназначен для работы с удаленными пользователями (до 4...7 км) и представляет собой витую пару. Его функционирование основано на использовании дуплексного режима (full-duplex), т. е. передачи потока по линии связи в обоих направлениях одновременно. Посредством S/T-интерфейса осуществляют разводку внутри офиса компании, либо квартиры с помощью двухпарного кабеля; при этом возможно параллельное подключение до восьми устройств. Для согласования U- и S/T-интерфейсов обычно используют сетевые оконечные блоки NT1 (Network Terminator).

Основное оборудование ISDN

Любая концепция должна базироваться на комплексе средств, посредством которых она может быть реализована. Технология ISDN не является исключением. К числу основных средств ISDN можно отнести:

- ISDN-станции (ISDN-коммутаторы);
- ISDN-терминалы (цифровые телефонные аппараты);
- внутренние адаптеры ISDN (мосты/маршрутизаторы) для подключения ПК к ISDN-сети;
- внешние устройства (блоки) для подключения ПК или ЛВС к ISDN-сети как альтернатива адаптерам;
- блоки Network Terminator;
- линии связи (интерфейсы PRI и BRI).

На рис. 4.7 показан пример конфигурации ISDN, где изображены три устройства, подсоединеные к коммутатору ISDN, находящемуся на центральной станции. Два из этих устройств совместимы с ISDN, поэтому их можно подключить к устройствам NT2 через контрольную точку S. Третье устройство (стандартный, неспециализированный для ISDN телефон) подключено к TA через контрольную точку R. Любое из этих устройств можно также подсоединить к устройству NT1/2, заменяющему оба устройства: NT1 и NT2.

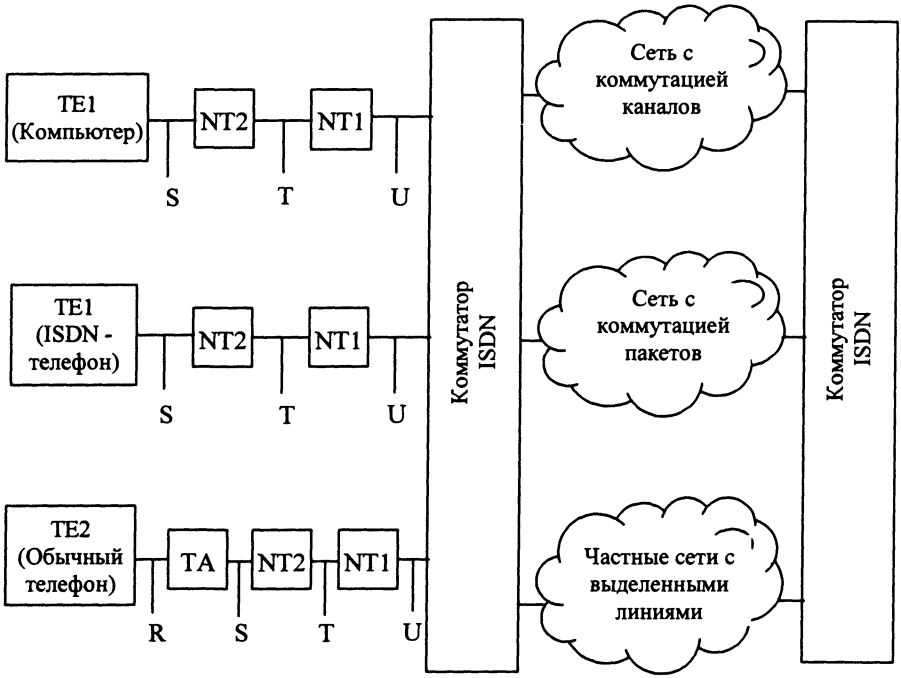


Рис. 4.7. Пример конфигурации ISDN

Уровни ISDN

Уровень 1 ISDN. В зависимости от того, является блок данных отправляемым за пределы терминала (из терминала в сеть) или входящим в пределы терминала (из сети в терминал) различают два вида блока данных физического уровня (Уровень 1) ISDN (рис. 4.8). Длина блоков данных равна 48 бит, 36 из которых представляют информацию. Биты F обеспечивают синхронизацию; L регулируют среднее значение бита; E разрешают конфликтные ситуации, когда несколько терминалов на какой-нибудь пассивной шине претендуют на один канал; A активирует устройства. Биты S еще не получили назначения. Биты B1, B2 и D предназначены для данных пользователя.

Физически к одной цепи можно подключить множество устройств пользователей ISDN. Для такой конфигурации столкновения могут быть результатом одновременной передачи двух терминалов. Поэтому ISDN предусматривает средства для определения конфликтов в канале связи. При получении устройством NT бита D из TE оно отражает этот бит эхо-сигналом обратно в соседнюю позицию бита E. Устройство TE ожидает, что соседний бит E должен быть тем же самым, что и бит D, который он передал в последней передаче.

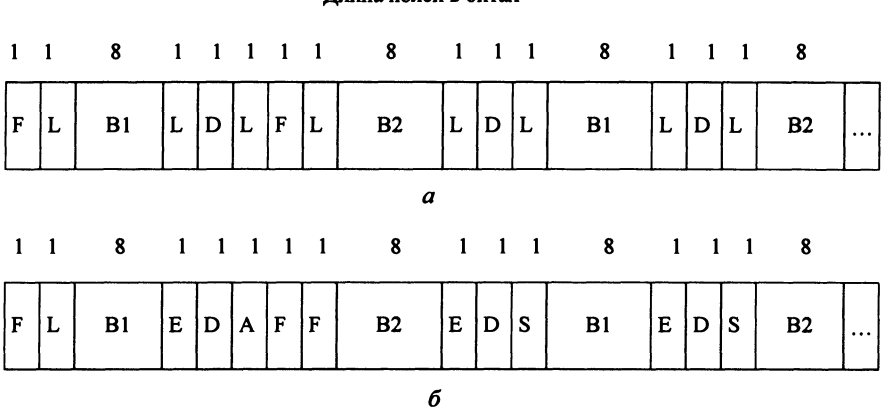


Рис. 4.8. Форматы блоков данных физического уровня ISDN:
а – NT-кадр (из сети в терминал); *б* – TE-кадр (из терминала в сеть), F – бит кадра, L – бит DC балансировки, E – бит эхо D-канала (эхо предыдущего D-бита), D – D-канал (4 бит × 4000 fps = 16 Kbps), A – бит активации, S – резерв, B1 – биты B в канале 1, B2 – биты B в канале 2

Уровень 2 ISDN. Данный уровень обеспечивается процедурой доступа к каналу связи, реализованной в виде протокола обмена сигналами ISDN к D-каналу (Link Access Procedure, D-channel), известной также как LAPD. Процедура LAPD аналогична процедуре управления каналом передачи данных высокого уровня (HDLC) и процедуре доступа к каналу связи, сбалансированной (LAPB). Процедуру LAPD используют в D-канале для обеспечения передачи и приема потока данных и соответствующей управляющей и сигнализирующей информации. Формат блока данных LAPD (рис.4.9) очень похож на формат HDLC. В нем так же, как в HDLC, использован супервизорный, информационный и ненумерованный блоки данных. Протокол LAPD формально определен в спецификациях CCITT Q.920 и CCITT Q.921.

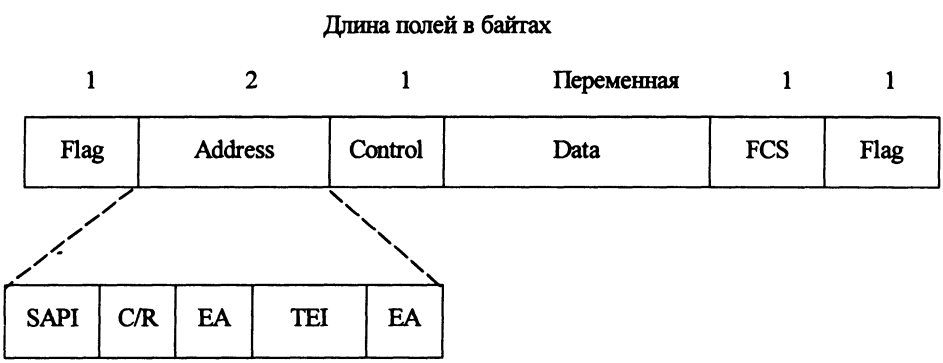


Рис. 4.9. Формат блока данных LAPD

Поля Flag и Control LAPD идентичны таким полям в HDLC. Длина поля Address LAPD может составлять 1 или 2 байт. Если в первом байте задан бит расширенного адреса (EA), то адрес состоит из одного байта; если он не задан, то адрес состоит из двух байтов. Первый байт адресного поля содержит идентификатор точки доступа к услугам (SAPI – Servise Access Point Identifier), определяющий главный вход, в котором услуги LAPD предоставляются Уровню 3. Бит C/R указывает, содержит ли блок данных команду или ответный сигнал. Поле идентификатора конечной точки терминала (TEI – Terminal Endpoint Identifier) указывает, является ли терминал единственным или их много. Этот идентификатор единственный из перечисленных выше, который указывает на широковещание.

Уровень 3 ISDN. Для передачи сигналов ISDN используют две спецификации Уровня 3: CCITT 1.450 (известная также как CCITT Q.930) и CCITT 1.451 (известная также как CCITT Q.931). Вместе оба протокола обеспечивают три типа соединения:

- пользователь – пользователь;
- с коммутацией каналов;
- с коммутацией пакетов.

В протоколах CCITT 1.450 и 1.451 определены разнообразные сообщения по организации и завершению обращения, информационные и смешанные сообщения, в том числе SETUP (УСТАНОВКА), CONNECT (ПОДКЛЮЧАТЬ), RELEASE (ОТКЛЮЧЕНИЕ), USER INFORMATION (ИНФОРМАЦИЯ ПОЛЬЗОВАТЕЛЯ), CANCEL (ОТМЕНА), STATUS (СОСТОЯНИЕ) и DISCONNECT (РАЗЪЕДИНЯТЬ). Эти сообщения функционально схожи с сообщениями, которые обеспечивает протокол X.25.

4.3. Технология Frame relay

Сети Frame relay – сравнительно новые сети, которые более удобны для передачи пульсирующего трафика локальных сетей по сравнению с сетями X.25, правда это преимущество проявляется лишь тогда, когда каналы связи приближаются по качеству к каналам локальных сетей, а для глобальных каналов такое качество обычно достижимо только при использовании волоконно-оптических кабелей. Преимущество сетей Frame relay заключено в их низкой протокольной избыточности и дейтаграммном режиме работы, что обеспечивает высокую пропускную способность и небольшие задержки кадров. Надежную передачу кадров технология Frame relay не обеспечивает. Сети Frame relay специально разрабатывались как общественные сети для соединения частных локальных сетей. Они обеспечивают скорость передачи данных до 2 Мбит/с.

Особенностью технологии Frame relay является гарантированная поддержка основных показателей качества транспортного обслуживания локальных сетей – средней скорости передачи данных по виртуальному каналу при допустимых пульсациях трафика. Кроме технологии Frame relay гарантии качества обслуживания на сегодня может предоставить только технология ATM, в то

время как остальные технологии предоставляют требуемое качество обслуживания на сегодня только в режиме «с максимальными усилиями» (best effort), т. е. без гарантий.

Стандарты Frame relay так же, как и X.25, определяют два типа виртуальных каналов: постоянные (PVC) и коммутируемые (SVC). Однако производители оборудования Frame relay и поставщики услуг сетей Frame relay начали с поддержки только постоянных виртуальных каналов. Это, естественно, является большим упрощением технологии. В последние годы разработано оборудование, поддерживающее коммутируемые виртуальные каналы, и соответственно появились поставщики, предлагающие такую услугу.

Стек протоколов Frame relay

Технология Frame relay использует для передачи данных принцип виртуальных соединений, аналогичный применяемому в сетях X.25. Отличие заключается в том, что стек протоколов Frame relay передает кадры (при установленном виртуальном соединении) по протоколам только физического и канального уровней, в то время как в сетях X.25 и после установления соединения пользовательские данные передаются протоколом сетевого уровня. Кроме того, протокол канального уровня LAP-F в сетях Frame relay имеет два режима работы: основной (core) и управляющий (control). В основном режиме кадры передаются без преобразования и контроля, как и в коммутаторах локальных сетей. За счет этого в сетях Frame relay высокая производительность, так как кадры в коммутаторах не подвергают преобразованию, а сеть не передает квитанции подтверждения между коммутаторами на каждый пользовательский кадр, как это происходит в сети X.25. Пульсации трафика передаются сетью Frame relay достаточно быстро и без больших задержек. При таком подходе накладные расходы при передаче пакетов локальных сетей меньше, так как они вкладываются сразу в кадры канального уровня, а не в пакеты сетевого уровня, как это происходит в сетях X.25.

Структура стека протоколов Frame relay представлена на рис.4.10. Как видим из рисунка сети Frame relay заимствуют многое из стека протоколов ISDN. Основу технологии Frame relay составляет протокол LAP-F core, который представляет собой упрощенную версию протокола LAPD. Протокол LAP-F (стандарт Q.922 ITU-T) работает на любых каналах сети ISDN, а также на каналах типа T1/E1. Терминальное оборудование посылает в сеть кадры LAP-F в любой момент времени, считая, что виртуальный канал в сети коммутаторов уже проложен. При использовании PVC оборудованию Frame relay нужно поддерживать только протокол LAP-F core.

Протокол LAP-F control является необязательной надстройкой над LAP-F core, которая выполняет функции контроля доставки кадров и управления потоком.

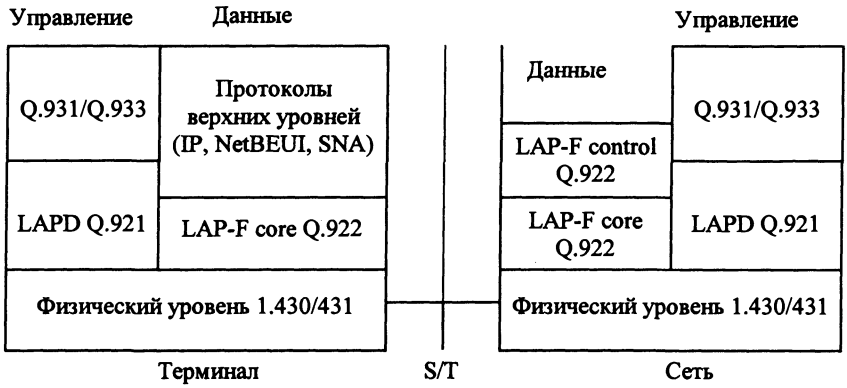


Рис. 4.10. Стек протоколов Frame relay

Для установки коммутируемых виртуальных каналов используется канал D пользовательского интерфейса с протоколом LAPD, который обеспечивает надежную передачу кадров в сетях ISDN. Поверх этого протокола работает протокол Q.931 или протокол Q.933 (упрощение и модификация протокола Q.931 ISDN), устанавливающий виртуальное соединение на основе адресов конечных абонентов (в стандарте E.164 или ISO 7498), а также номера виртуального соединения, который в технологии Frame relay называют DLCI – Data Link Connection Identifier.

После того как коммутируемый виртуальный канал в сети Frame relay установлен посредством протоколов LAPD и Q.931/933, кадры можно транслировать по протоколу LAP-F. Этот протокол коммутирует их с помощью таблиц коммутации портов, в которых использованы локальные значения DLCI. Протокол LAP-F core выполняет не все функции канального уровня по сравнению с протоколом LAPD, поэтому ITU-T изображает его на пол-уровня ниже, чем протокол LAPD, оставляя место для функций надежной передачи пакетов протоколу LAP-F control.

Из-за того, что технология Frame relay ограничена канальным уровнем, она хорошо согласуется с идеей инкапсуляции пакетов единого сетевого протокола, например IP, в кадры канального уровня любых сетей, составляющих интерсеть. Процедуры взаимодействия протоколов сетевого уровня с технологией Frame relay стандартизованы. Принята спецификация RFC 1490, определяющая методы инкапсуляции в трафик Frame relay графика сетевых протоколов и протоколов канального уровня локальных сетей и SNA.

Особенностью технологии Frame relay является отказ от коррекции обнаруженных в кадрах искажений. Протокол Frame relay подразумевает, что конечные узлы будут обнаруживать и корректировать ошибки за счет работы протоколов транспортного или более высоких уровней. Это требует некоторой степени интеллектуальности от конечного оборудования. В современных локальных сетях данное требование, как правило, выполняется. В этом отношении техноло-

гия Frame relay близка к технологиям локальных сетей, таким как Ethernet, Token Ring и FDDI, которые тоже только отбрасывают искаженные кадры и повторно их не передают.

Структура кадра протокола LAP-F

Структура кадра протокола LAP-F приведена на рис. 4.11. За основу структуры взят формат кадра HDLC, но в поле адреса существенно изменен формат, а поле управления отсутствует.

Поле номера виртуального соединения (DLCI) состоит из 10 бит, что позволяет использовать до 1024 виртуальных соединений. Поле DLCI может занимать и большее число разрядов, этим управляют признаки EA0 и EA1 (Extended Address – расширенный адрес). Если бит в этом признаке установлен в «0», то признак называется EA0 и означает, что в следующем байте имеется продолжение поля адреса, а если бит признака равен «1», то поле называется EA1 и оно определяет окончание поля адреса.

Десятиразрядный формат DLCI является основным, но при использовании 3 байт для адресации поле DLCI имеет длину 16 бит, а при использовании 4 байт – 23 бит.

Стандарты Frame relay (ANSI, ITU-T) распределяют адреса DLCI между пользователями и сетью следующим образом:

- 0 – используется для виртуального канала локального управления (LMI);
- 1–15 – зарезервированы для дальнейшего применения;
- 16–991 – используют абоненты для нумерации PVC и SVC;
- 992–1007 – использует сетевая транспортная служба для внутрисетевых соединений;
- 1008–1022 – зарезервированы для дальнейшего применения;
- 1023 – используют для управления канальным уровнем.

Таким образом, в любом интерфейсе Frame relay для оконечных устройств пользователя отводится 976 адресов DLCI. Поле данных может иметь размер до 4056 байт.

Поле C/R имеет обычный для протокола семейства HDLC смысл – признак «команда-ответ». Протоколом это поле не используется и передается по сети прозрачно.

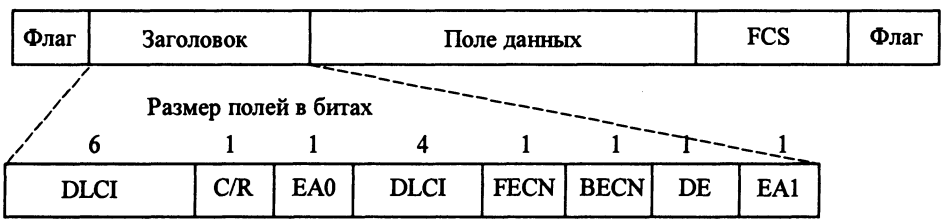


Рис. 4.11. Формат кадра LAP-F

Поля FECN, BECN и DE используются протоколом для управлением трафиком и поддержания заданного качества обслуживания виртуального канала. FECN – информирует узел назначения о заторе, BECN – информирует узел-источник о заторе, DE – идентифицирует кадры, которые могут быть сброшены в случае затора.

Качество обслуживания

Для каждого виртуального соединения определено несколько параметров, влияющих на качество обслуживания:

- *CIR* (Committed Information Rate) – согласованная информационная скорость, с которой сеть будет передавать данные пользователя;
- B_c (Committed Burst Size) – согласованный объем пульсации, т. е. максимальное количество байтов, которое сеть будет передавать от пользователя за интервал времени T;
- B_e (Excess Burst Size) – дополнительный объем пульсации, т. е. максимальное количество байтов, которое сеть будет пытаться передать сверх установленного значения B_c за интервал времени T.

Если эти величины известны, то время можно вычислить по формуле:

$$T = B_c/CIR.$$

Можно задать значения *CIR* и T, тогда производной величиной станет величина всплеска трафика B_c. Реакция сети на поведение пользователя приведена на рис. 4.12.

Гарантий по задержкам передачи кадров технология Frame relay не дает, оставляя эту услугу сетям ATM.

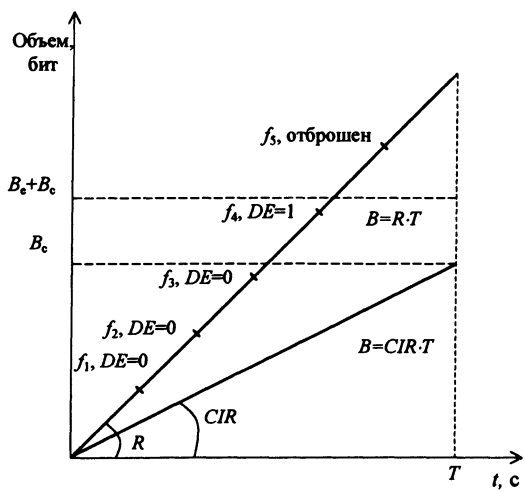


Рис. 4.12. Реакция сети на поведение пользователя:
R – скорость канала доступа; f_1–f_5 – кадры

Основным параметром, по которому абонент и сеть заключают соглашение при установлении виртуального соединения, является согласованная скорость передачи данных. Для постоянных виртуальных каналов это соглашение определяется контрактом на пользование услугами сети. При установлении коммутируемого виртуального канала соглашение о качестве обслуживания заключается автоматически с помощью протокола Q.931/933. Требуемые параметры *CIR*, B_c и B_e передаются в пакете запроса на установление соединения.

Так как скорость передачи данных измеряют на каком-то интервале времени, то интервал T и является тем контрольным интервалом, на котором проверяют условия соглашения. В общем случае пользователь не должен за этот интервал передать в сеть данные со средней скоростью, превосходящей *CIR*. Если же он нарушает соглашение, то сеть не только не гарантирует доставку кадра, но помечает этот кадр признаком *DE* (Discard Eligibility), равным «1», т. е. как кадр, подлежащий удалению. Однако кадры, отмеченные таким признаком, удаляются из сети только в том случае, если коммутаторы сети испытывают перегрузки. Если же перегрузок нет, то кадры с признаком $DE = 1$ доставляются адресату. Такое щадящее поведение сети соответствует случаю, когда общее количество данных, переданных пользователем в сеть за период T, не превышает объема $B_c + B_e$. Если же этот порог превышен, то кадр не помечается признаком *DE*, а немедленно удаляется из сети.

На рис.4.12 изображен случай, когда за интервал времени T в сеть по виртуальному каналу поступило 5 кадров. Средняя скорость поступления информации в сеть составила на этом интервале R бит/с, и она оказалась выше *CIR*. Кадры f_1, f_2 и f_3 доставили в сеть данные, суммарный объем которых не превысил порог B_c, поэтому эти кадры передаются дальше транзитом с признаком $DE = 0$. Данные кадра f_4, прибавленные к данным кадров f_1, f_2 и f_3, уже превысили порог B_c, но еще не превысили порога $B_c + B_e$, поэтому кадр f_4 также передается дальше, но уже с признаком $DE = 1$. Данные кадра f_5, прибавленные к данным предыдущих кадров, превысили порог $B_c + B_e$, поэтому этот кадр был удален из сети.

Для контроля соглашения о параметрах качества обслуживания все коммутаторы сети Frame relay выполняют так называемый алгоритм «дырявого ведра» (Leaky Bucket). Алгоритм использует счетчик поступивших от пользователя байт. Каждые T секунд этот счетчик уменьшает свое значение на величину B_c (или же сбрасывается в «0», если значение счетчика меньше, чем B_c). Все кадры, данные которых не увеличили значение счетчика свыше порога B_c, проходят в сеть со значением признака $DE = 0$. Кадры, данные которых привели к значению счетчика, большему B_c, но меньшему $B_c + B_e$, также передаются в сеть, но с признаком $DE = 1$. И, наконец, кадры, которые привели к значению счетчика, большему $B_c + B_e$, коммутатор отбрасывает.

Пользователь может заказать включение не всех параметров качества обслуживания на данном виртуальном канале, а только некоторых. Например,

можно использовать только параметры *CIR* и B_c. Этот вариант дает более качественное обслуживание, так как коммутатор никогда не отбрасывает кадры сразу. Он только помечает кадры, данные которых превышают порог B_c за время T, признаком $DE = 1$. Если в сети не наблюдаются перегрузки, то кадры такого канала всегда доходят до конечного узла, даже если пользователь постоянно нарушает договор с сетью.

Популярен еще один вид заказа на качество обслуживания, при котором оговаривается только порог B_e, а скорость *CIR* полагают равной нулю. Все кадры такого канала сразу же отмечают признаком $DE = 1$, но отправляют в сеть, а при превышении порога B_e их отбрасывают. Контрольный интервал времени в этом случае равен

$$T = B_e/R, \tag{4.2}$$

где R – скорость доступа канала.

На рис. 4.13. приведен пример сети Frame relay с пятью удаленными отделениями. Обычно доступ к сети осуществляют каналы с большей, чем у *CIR*, пропускной способностью. Но при этом пользователь платит не за пропускную способность канала, а за заказанные параметры *CIR*, B_c и B_e. Так, при использовании в качестве канала доступа канала T1 и заказа службы со скоростью *CIR*, равной 128 кбит/с, пользователь будет платить только за скорость 128 кбит/с, а скорость канала T1 в 1,544 Мбит/с будет влиять на верхнюю границу возможной пульсации $B_c + B_e$.

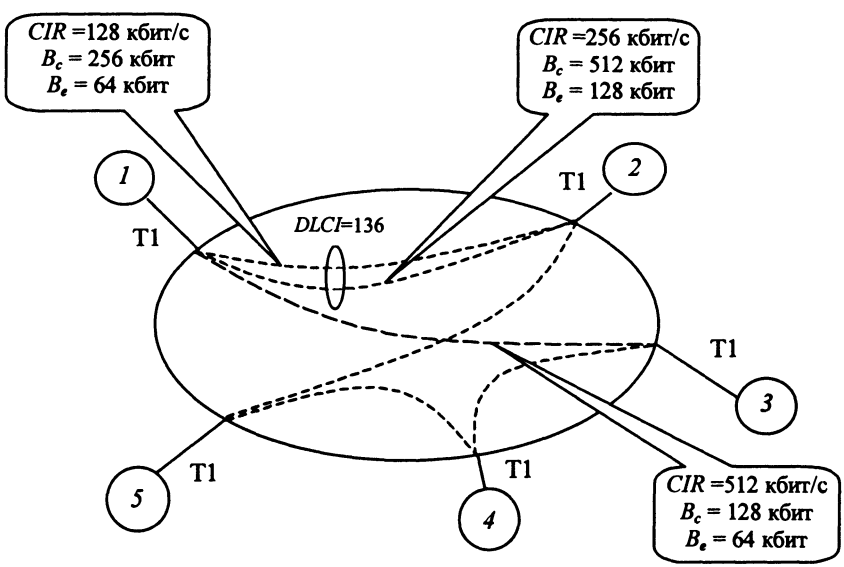


Рис. 4.13. Пример использования сети Frame relay

Параметры качества обслуживания могут быть различными для разных направлений виртуального канала. Так, на рис. 4.13 абонент *1* соединен с абонентом *2* виртуальным каналом с DLCI = 136. При направлении от абонента *1* к абоненту *2* канал имеет среднюю скорость 128 кбит/с с пульсациями B_c = 256 кбит (интервал T составил 1 с) и B_e = 64 кбит. А при передаче кадров в обратном направлении средняя скорость уже может достигать 256 кбит/с с пульсациями B_c = 512 кбит и B_e = 128 кбит.

Механизм заказа средней пропускной способности и максимальной пульсации является основным механизмом управления потоками кадров в сетях Frame relay. Соглашения должны заключаться таким образом, чтобы сумма средних скоростей виртуальных каналов не превосходила возможностей портов коммутаторов. При заказе постоянных каналов за это отвечает администратор, а при установлении коммутируемых виртуальных каналов – программное обеспечение коммутаторов. При правильно взятых на себя обязательствах сеть борется с перегрузками путем удаления кадров с признаком DE = 1 и кадров, превысивших порог $B_c + B_e$.

Управление перегрузками

В технологии Frame relay определен еще и дополнительный (необязательный) механизм управления кадрами – механизм оповещения конечных пользователей о том, что в коммутаторах сети возникли перегрузки (переполнение необработанными кадрами). Бит FECN (Forward Explicit Congestion Bit) кадра извещает об этом принимающую сторону. На основании значения этого бита принимающая сторона должна с помощью протоколов более высоких уровней (TCP/IP, SPX и т. п.) известить передающую сторону о том, что та должна снизить интенсивность отправки пакетов в сеть.

Бит BECN (Backward Explicit Congestion Bit) извещает о переполнении передающую сторону и является требованием немедленного снижения темпа передачи. Бит BECN обычно отрабатывается на уровне устройств доступа к сети Frame relay – маршрутизаторов, мультиплексоров и устройств *CSU/DSU*. Протокол Frame relay не требует от устройств, получивших кадры с установленными битами FECN и BECN, немедленного прекращения передачи кадров в данном направлении, как того требуют кадры RNR сетей X.25. Эти биты служат указанием для протоколов более высоких уровней (TCP, SPX, NCP и т. п.) о снижении темпа передачи пакетов. Так как регулирование потока инициируется в различных протоколах по-разному – как принимающей стороной, так и передающей, – то разработчики протоколов Frame relay учли оба направления снабжения предупреждающей информацией о переполнении сети.

В общем случае биты FECN и BECN могут игнорироваться. Но обычно устройства доступа к сети Frame relay (FRAD – Frame relay Access Device) отрабатывают по крайней мере признак BECN.

Создание коммутируемого виртуального канала

При создании коммутируемого виртуального канала параметры качества обслуживания передаются в сеть с помощью протокола Q.931. Этот протокол устанавливает виртуальное соединение с помощью нескольких служебных пакетов.

Абонент сети Frame relay, желающий установить коммутируемое виртуальное соединение с другим абонентом, должен передать в сеть по каналу D сообщение SETUP, которое имеет несколько параметров, в том числе:

- DLCI;
- адрес назначения (в формате E.164, X.121 или ISO 7498);
- максимальный размер кадра в данном виртуальном соединении;
- запрашиваемое значение *CIR* для двух направлений;
- запрашиваемое значение B_c для двух направлений;
- запрашиваемое значение B_e для двух направлений.

Коммутатор, с которым соединен пользователь, сразу же передает пользователю пакет CALL PROCEEDING (вызов обрабатывается). Затем он анализирует параметры, указанные в пакете, и если коммутатор может их удовлетворить (располагая, естественно, информацией о том, какие виртуальные каналы на каждом порту он уже поддерживает), то пересылает сообщение SETUP следующему коммутатору, который выбирается по таблице маршрутизации. Протокол автоматического составления таблиц маршрутизации для технологии Frame relay не стандартизирован, поэтому можно использовать фирменный протокол производителя оборудования или же ручное составление таблицы. Если все коммутаторы на пути к конечному узлу согласны принять запрос, то пакет SETUP передается в конечном счете вызываемому абоненту. Вызываемый абонент немедленно передает в сеть пакет CALL PROCEEDING и начинает обрабатывать запрос. Если запрос принят, то вызываемый абонент передает в сеть новый пакет – CONNECT, который проходит в обратном порядке по виртуальному пути. Все коммутаторы должны отметить, что данный виртуальный канал принят вызываемым абонентом. При поступлении сообщения CONNECT вызывающему абоненту он должен передать в сеть пакет CONNECT ACKNOWLEDGE.

Сеть также должна передать вызываемому абоненту пакет CONNECT ACKNOWLEDGE. На этом соединение считается установленным и по виртуальному каналу можно передавать данные.

Использование сетей Frame relay

Полезная пропускная способность прикладных протоколов при работе через сети Frame relay зависит от качества каналов и методов восстановления пакетов на уровнях стека, расположенного над протоколом Frame relay. Поэтому сети Frame relay следует применять только на магистральных каналах с волоконно-оптическим кабелем высокого качества. Каналы доступа используют витую пару, как это разрешает интерфейс G.703, или абонентское окончание

ISDN. Применяемая на каналах доступа аппаратура передачи данных должна обеспечивать приемлемый уровень искажения данных не ниже 10^{-6}.

На величины задержек сеть Frame relay гарантий не дает, что является основной причиной, сдерживающей применение этих сетей для передачи голоса. Другим отличием сетей Frame relay от ATM – является низкая скорость доступа – 2 Мбит/с, что для передачи видео часто недостаточно. Тем не менее, многие производители оборудования для сетей Frame relay поддерживают передачу голоса, что обеспечивается присвоением кадрам, переносящим замеры голоса, приоритетов. Магистральные коммутаторы Frame relay должны обслуживать приоритетные кадры в первую очередь. Кроме того, желательно, чтобы сеть Frame relay, передающая кадры с замерами голоса, была недогруженной. При этом в коммутаторах не возникают очереди кадров, и средние задержки в очередях близки к нулевым.

Для качественной передачи голоса необходимо также соблюдение еще одного условия – передавать замеры голоса только в кадрах небольших размеров, иначе на качество будут влиять задержки упаковки замеров в кадр, так называемые задержки пакетизации, которые более подробно рассматриваются в разделе, посвященном технологии ATM.

При использовании PVC сеть Frame relay удобна для объединенных локальных сетей с помощью мостов, так как в этом случае от моста не нужна поддержка механизма установления виртуального канала. Мост может отправлять кадры протокола Ethernet или FDDI непосредственно в кадрах LAP-F или же использовать поверх протокола LAP-F протокол PPP. Стандарт Internet RFC 1490 определяет формат заголовка SNAP для случая передачи через сеть Frame relay непосредственно кадров канального уровня.

Чаще доступ к сетям Frame relay реализуют не удаленные мосты, а маршрутизаторы, поддерживающие на последовательных портах протокол Frame relay и называемые устройствами доступа FRAD.

Виртуальные каналы в качестве основы построения корпоративной сети имеют один недостаток – при большом количестве точек доступа и смешанном характере связей необходимо большое число виртуальных каналов, каждый из которых оплачивается отдельно. В сетях с маршрутизацией отдельных пакетов, таких как TCP/IP, абонент платит только за количество точек доступа, а не за количество связей между ними.

4.4. Технология ATM

Технология передачи данных *A*TM (Asynchronous Transfer Mode – режим асинхронной пересылки) была разработана и специфицирована при проектировании ISDN (Integrated Services Digital Network – цифровая сеть интегрированных служб) отделом коммуникаций Международного союза по электросвязи (ITU-T – International Telecommunications Union – Telecommunications).

Технология ATM специально ориентирована на работу с информацией различного типа:

- речевым трафиком, традиционно обслуживаемым телефонными сетями;
- трафиком данных, который обычно передается по компьютерным сетям;
- трафиком мультимедиа, сочетающим в себе статические изображения, аудио- и видеоинформацию.

Технология ATM реализует коммутацию, ориентированную на аппаратуру, программные средства которой обеспечивают безукоризненное выполнение соединений с неограниченной полосой пропускания для передачи данных, видео- и голосовых сообщений. Технология пакетной коммутации ATM применяет короткие пакеты фиксированной длины, называемые ячейками (cell). Ячейка ATM имеет размер 53 байт, пять из которых составляют заголовок, оставшиеся 48 – собственно информацию (рис. 4.14). Так как ячейки имеют фиксированную длину, конструкция ATM-коммутатора более проста, задержки при обработке данных сокращены, дисперсия задержек снижена, что существенно для таких чувствительных к задержкам видам коммуникационного обслуживания, как передача голосовых сообщений и видео.

Поле *управление потоком* (GFC – Generic Flow Control) длиной 4 бит используют только при взаимодействии конечного узла и первого коммутатора сети ATM.

Поле *идентификатор виртуального пути* (VPI – Virtual Path Identifier) длиной 12 бит используют для группирования виртуальных каналов с целью маршрутизации.

Поле *идентификатор виртуального канала* (VCI – Virtual Channel Identifier) –16-разрядное, идентифицирует конкретный виртуальный канал в виртуальном пути.

Поле тип информационного наполнения (PT – Payload Type) длиной 3 бит, идентифицирует тип данных, содержащихся в поле информационного наполнения. Кроме того, 1 бит этого поля используется для указания перегрузки в сети.

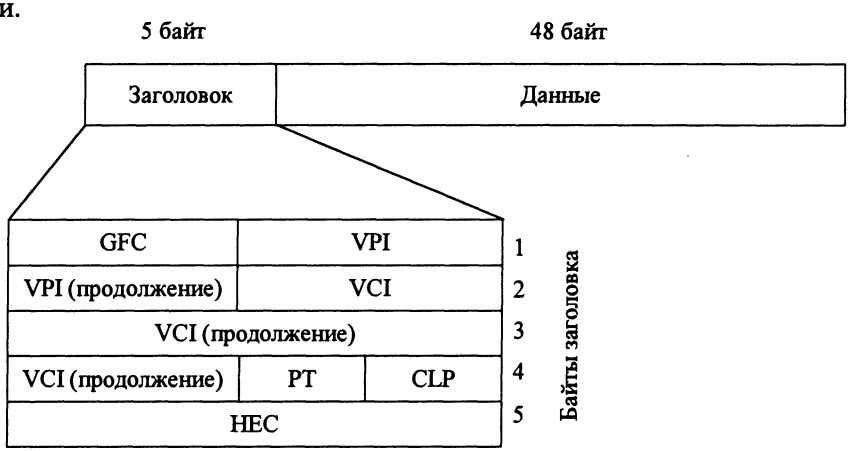


Рис. 4.14. Формат ячейки ATM

Однобитовое поле *приоритет потери ячейки* (CLP – Cell Loss Priority) позволяет оборудованию ATM определить, какие ячейки нужно отбрасывать в первую очередь при возникновении перегрузки. Ячейки с CLP = 1 являются для сети низкоприоритетными, а ячейки с CLP = 0 – высокоприоритетными.

Поле контроля ошибок заголовка (HEC – Header Error Check) содержит значение кода обнаружения и коррекции ошибок. Его иногда используют для исправления ошибок в пяти октетах (40 бит) заголовка ячейки.

Поле данных ячейки содержит 48 октетов (384 бит) данных пользователя и/или дополнительной управляющей информации.

В заголовке ATM виртуальный канал обозначен комбинацией двух полей – VPI (идентификатор виртуального пути) и VCI (идентификатор виртуального канала). Виртуальный путь используют в случаях, когда два пользователя ATM имеют свои собственные коммутаторы на каждом конце пути и, следовательно, могут организовывать и поддерживать свои виртуальные соединения. Виртуальный путь напоминает канал, содержащий множество кабелей, по каждому из которых может быть организовано виртуальное соединение.

Поскольку виртуальные устройства подобны реальным, они также бывают выделенные или коммутируемые. В сетях ATM «выделенные» соединения называют постоянными виртуальными устройствами (PVC), создаваемыми по соглашению между пользователем и оператором (подобно выделенной телефонной линии). Коммутируемые соединения ATM используют коммутируемые виртуальные устройства (SVC), устанавливаемые путем передачи специальных сигналов между пользователем и сетью. Протокол, используемый ATM для управления виртуальными устройствами подобен протоколу ISDN. Вариант протокола для ISDN описан в стандарте Q.931, ATM – в Q.2931.

Существуют два формата заголовка ячейки: интерфейс «абонент-сеть» (UNI) и интерфейс «узел-сеть» (NNI). Интерфейс UNI предназначан, в первую очередь, для узлов оконечных пользователей, таких, как рабочие станции. Интерфейс NNI разработан с целью стандартизации метода установления соединений между маршрутизаторами и коммутаторами.

Сети ATM состоят из трех основных компонентов (рис. 4.15):

- коммутаторы ATM (switch);
- конечные точки ATM (ES);
- маршруты пересылки (TP).

Коммутаторы *ATM* выполняют функции, связанные с маршрутизацией информации от пользователя-отправителя к пользователю-получателю, их называют промежуточными системами (IS, intermediate path). Различают две основные категории коммутаторов:

общественные ATM (public ATM switch);

частные ATM (private ATM switch).

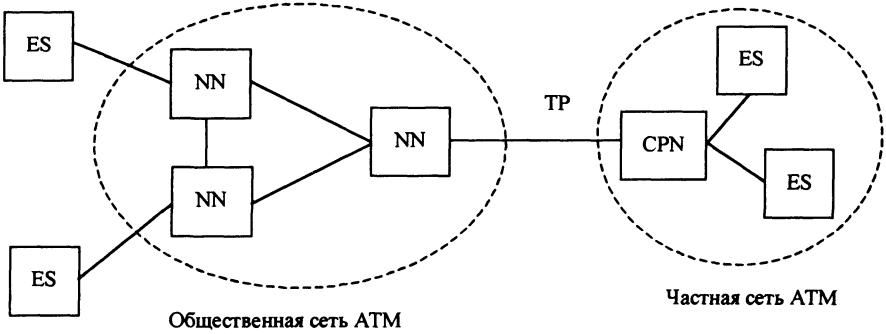


Рис. 4.15. Структура сети ATM

Существуют коммутаторы, способные выполнять функции как частных, так и общественных коммутаторов. Общественный коммутатор ATM – это часть общественной сети поставщика телекоммуникационных услуг, по стандартам ATM такие коммутаторы называются *сетевыми узлами* (NN – Network Node). Частный коммутатор ATM принадлежит и обслуживается организацией-пользователем, в стандартах ATM он называется *узлом на территории потребителя* (CPN – Customer Premises Node). Такие коммутаторы поставляются в основном производителями сетевых плат, концентраторов, мостов и маршрутизаторов.

Конечные точки сетей ATM – это устройства, которые могут служить отправителем или получателем пользовательских данных. Конечные точки ATM (или, иначе, оконечные системы ES – End Systems) соединены непосредственно с общественным или частным коммутатором ATM. Конечной точкой ATM может быть обычная компьютерная система с платой сетевого интерфейса ATM и соответствующее программное обеспечение, или специальное сетевое устройство ATM, к которому через обычные адаптеры локальной сети подключены несколько вычислительных систем. Взаимодействие между конечными точками ATM и коммутаторами осуществляется через коммуникационные связи, называемые путями или маршрутами пересылки (TP – Transmission Path).

Маршруты пересылки данных можно реализовывать с помощью различных типов коммуникационных сетей: на основе волоконно-оптических или электрических носителей информации.

Уровни и классы служб ATM

Разработчики архитектуры ATM разделили операции, выполняемые в сети устройствами ATM, на три функциональных уровня (рис. 4.16).

Физический уровень отправляет и принимает информацию в виде электрических или оптических сигналов, передаваемых по физическому пути передачи данных. Эта функция физических коммуникаций предусматривает преобразование ячеек в непрерывный поток битов или кадров и обратно, а также предполагает использование различных видов (физического) кодирования и декодирования данных, содержащихся в каждой из ячеек.

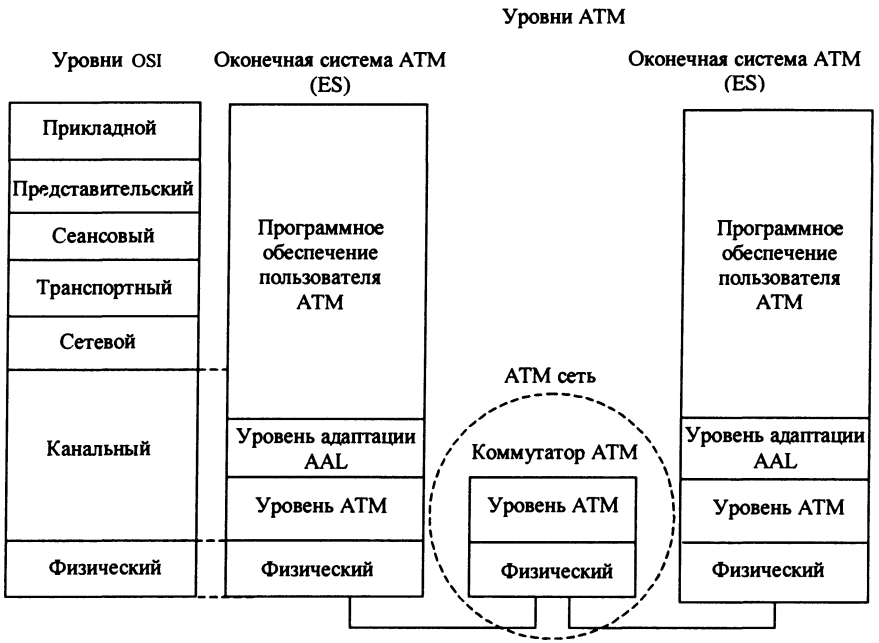


Рис. 4.16. Связь уровней ATM и OSI/ISO

Основной функцией уровня ATM является коммутация ячеек. На этом уровне устройство ATM принимает ячейки, поступающие по пути передачи данных, определяет путь дальнейшей пересылки, по которому эти ячейки следует ретранслировать, и форматирует заголовок каждой отправляемой ячейки.

Уровень адаптации ATM (AAL) предлагает интерфейс между сетью ATM и пользовательским программным обеспечением ATM, обычно реализуемым в подсистеме сетевого ПО. AAL применяется только в оконечных устройствах, но не в коммутаторах ATM. В оконечном устройстве, играющем роль отправителя, AAL принимает битовый поток от пользовательского программного обеспечения ATM и структурирует его в виде ячеек, подходящих для транспортировки по сети ATM. В принимающем оконечном устройстве ATM соответствующий уровень AAL получает ячейки из сети, воссоздает исходный битовый поток и передает его принимающему пользовательскому ПО.

Классы служб AAL. Так как ATM может передавать сетевой трафик различных типов, то для уровня адаптации ATM определены четыре класса служб (*A*, *B*, *C* и *D*), приведенные в табл. 4.1.

Класс A (эмуляция цепей). Служба класса *A* ориентирована на создание соединения и поддерживает постоянную скорость передачи битов и временные соотношения между отправлением и получением информации. Таким образом, адресат получает поток данных с постоянной скоростью (с какой ее передает отправитель). Службу этого класса можно использовать для передачи аудио- и видеоданных вместо обычной телекоммуникационной связи с коммутацией цепей (т. е. вместо аналогового канала).

Таблица 4.1. **Классы служб для уровня AAL**

Класс служб	Требования к параметрам потоков данных						Тип сообщения
	Синхронизация		Скорость передачи		Установление соединения		
	да	нет	постоянная	переменная	да	нет	
A	+	–	+	–	+	–	Видеоинформация
B	+	–	–	+	+	–	Аудио- и видеоинформация
C	–	+	–	+	+	–	Цифровые данные
D	–	+	–	+	–	+	Цифровые данные

Класс B (переменная скорость передачи). Данная служба во многом аналогична службе класса *A*, она ориентирована на соединение, имеет переменную скорость передачи и поддерживает временные соотношения между отправлением и получением данных. Кроме того, эта служба предусматривает передачу уплотненной (сжатой) аудио- и видеоинформации и может использоваться, например, в видеоконференциях, где при ограниченных задержках изменяющаяся скорость передачи данных считается допустимой.

Класс C (передача данных, ориентированная на создание соединения). Служба класса *C* также ориентирована на создание соединения, но не поддерживает временные соотношения. Служба этого класса требует создания двухточечного соединения между отправителем и получателем. Пользователь-отправитель передает информацию в сеть в виде пакетов переменного размера, которые получает целевое программное обеспечение пользователя ATM. Поступающие к целевому пользователю пакеты могут приходить с отличной от исходной скоростью. Эта служба обеспечивает обмен данными подобно обычным компьютерным сетям.

Класс D (передача данных, не ориентированная на создание соединения). Служба класса *D* не ориентирована на соединение, она предназначена для передачи данных без создания соединения, что обычно применяют в локальных сетях. В этой службе информация передается по сети ATM в виде пакетов переменного размера, которые поступают к получателю с переменной скоростью. При этом установление соединения не требуется. Каждый передаваемый пакет содержит полные адреса отправителя и получателя. Пакеты могут адресоваться как одному получателю, так и нескольким одновременно (многоадресная рассылка).

ATM и межсетевое взаимодействие

Существуют три варианта включения ATM в архитектуру межсетевого взаимодействия для современных и будущих приложений:

- эмуляция традиционных протоколов ЛВС с использованием оборудования ATM. В этом случае существующие приложения будут продолжать работать как раньше, а ATM-добавит к существующим протоколам новые, специально разработанные для приложений мультимедиа;
- подключение сервиса ATM напрямую к интерфейсам прикладных программ, используемых сегодня, в обход традиционных протоколов нижних уровней. Для поддержки этого варианта потребуется разработка новых API;
- использование новых API для новых приложений и эмуляция традиционных протоколов для существующих приложений.

Для обеспечения совместимости традиционных протоколов и оборудования локальных сетей с технологией ATM в ATM Forum была разработана спецификация, называемая LANE (LAN Emulation – эмуляция локальных сетей). Эта спецификация обеспечивает совместную работу сетей Ethernet и ATM на канальном уровне. При этом коммутаторы ATM работают в качестве высокоскоростных коммутаторов магистрали локальной сети, обеспечивая не только скорость, но и гибкость соединений коммутаторов ATM между собой, поддерживающих не только древовидные структуры, но и произвольную топологию связей.

В спецификации LANE определен способ преобразования кадров и MAC-адресов технологии Ethernet в ячейки и виртуальные каналы технологии ATM, а также и способ их обратного преобразования. Все действия по преобразованию протоколов выполняют специальные устройства, встраиваемые в обычные коммутаторы Ethernet. Именно поэтому ни коммутаторы ATM, ни рабочие станции сети Ethernet не замечают того, что работа осуществляется с чуждой им технологией. Такая прозрачность была одной из главных целей разработчиков спецификации LANE.

Так как в этой спецификации определяется только канальный уровень взаимодействия, то с помощью коммутаторов ATM и компонентов эмуляции можно образовывать только виртуальные сети, называемые в спецификации LANE эмулируемыми сетями, а для их соединения нужно использовать обычные маршрутизаторы.

Основными элементами, реализующими спецификацию LANE, являются программные компоненты LEC (LAN Emulation Client) и LES (LAN Emulation Server). Клиент LEC выполняет роль пограничного элемента, работающего между сетью ATM и станциями некоторой локальной сети (Ethernet). На каждую присоединенную к сети ATM локальную сеть приходится один клиент LEC. Таким клиентом обычно является ATM-LAN-конвертер, имеющий ATM-порт, с помощью которого он подключается к ATM-сети, а также сколько портов для подключения локальных сетей технологии Ethernet. Конвертор ATM-LAN имеет ATM-адрес для взаимодействия с другими конверторами по сети ATM.

Кроме того, он должен иметь информацию о MAC-адресах всех узлов каждой из локальных сетей, которые он присоединяет к сети ATM.

Сервер LES ведет общую таблицу соответствия MAC-адресов станций локальных сетей и ATM-адресов пограничных устройств с установленными на них компонентами LEC, к которым присоединены локальные сети, содержащие эти станции.

Таким образом, для каждой присоединенной сети сервер LES хранит один ATM-адрес пограничного устройства LEC и несколько MAC-адресов станций, входящих в эту сеть. Клиентские части LEC динамически регистрируют на сервере LES MAC-адреса каждой станции, заново подключаемой к присоединенной локальной сети.

Программные компоненты LEC и LES могут быть реализованы в любых устройствах – коммутаторах, маршрутизаторах, рабочих станциях ATM. Наиболее подходящим устройством для выполнения этих функций является коммутатор локальной сети Ethernet (ATM-LAN коммутатор), так как в нем есть таблица MAC-адресов всех устройств сети, которые обмениваются через него данными.

Когда элементу LEC требуется послать пакет данных через сеть ATM на станцию в другой локальной сети, также присоединенной к сети ATM, он посылает запрос на установление соответствия между MAC-адресом и ATM-адресом серверу LES. Сервер LES отвечает на запрос, указывая ATM-адрес пограничного устройства LEC, к которому присоединена сеть, содержащая станцию назначения. Зная ATM-адрес, устройство LEC исходной сети самостоятельно устанавливает виртуальное соединение через сеть ATM обычным способом. После установления связи кадры локальной сети преобразуются в ячейки ATM каждым элементом LEC с помощью стандартных функций сборки-разборки пакетов стека протоколов ATM. Кадр MAC помещается в область данных последовательности, состоящей из нескольких ячеек ATM.

Устройство LEC – получатель кадров MAC – производит сборку ячеек и выделение из них кадров MAC, которые и направляют в локальную сеть, где находится узел-получатель.

В каждый ATM-LAN-коммутатор встроен протокол LANE, в задачу которого входит передача принятого коммутатором MAC-кадра через ATM-сеть другому ATM-LAN-коммутатору. Так как к ATM-сети может быть подключено несколько локальных сетей, то при получении из локальной сети кадра с MAC-адресом назначения, ATM-LAN-коммутатор должен решить, к какому из остальных ATM-LAN-коммутаторов относится данный MAC-адрес.

Таким образом, коммутатор, принимая решение о передаче кадра, оперирует с двумя таблицами: локальной и транзитной.

Локальная таблица устанавливает соответствие MAC-адресов его локальной сети локальным портам. Транзитная таблица содержит для каждого MAC-адреса составной сети ATM-адрес пограничного коммутатора.

Спецификация LANE не определяет конкретный вид таблиц ATM-LAN-конверторов. Один из возможных вариантов этих таблиц приведен в табл. 4.2. и 4.3.

Таблица 4.2. **Таблица локальных адресов**

MAC-адрес	Номер порта
MAC1	1
MAC2	1
MAC3	2
MAC4	3
...	...

Таблица 4.3. **Таблица транзитных адресов**

MAC-адрес	Номер порта	ATM-адрес
MAC120	25	ATM1
MAC121	25	ATM1
MAC123	25	ATM2
MAC135	25	ATM3
...	...	...

Если ATM-LAN-коммутатор в результате просмотра адресных таблиц обнаруживает, что кадр нужно передать через ATM-сеть другому ATM-LAN-коммутатору, то он с помощью стека протоколов ATM устанавливает виртуальное соединение (VCC – Virtual Channel Connection) с этим коммутатором, а затем передает по нему кадр в форме потока ячеек ATM.

Важной задачей сервера эмуляции LES является автоматическое построение транзитных адресных таблиц ATM-LAN-коммутатора. Поскольку сети ATM, как и большинство территориальных сетей, не поддерживают широковещательность, то обнаружить через сеть ATM пограничные коммутаторы с помощью широковещательных запросов (как это делают, например, клиенты и серверы сетей NetWare) невозможно. Ручное задание ATM-адресов пограничных коммутаторов может оказаться обременительным занятием для администратора, если таких коммутаторов много и их набор часто претерпевает изменения, что характерно для локальных сетей.

Для автоматического построения транзитных адресных таблиц спецификация LANE предлагает использовать централизованный подход, т. е. возложить решение этой задачи на сервер LES. При своей инициализации LEC (ATM-LAN-коммутатор) сообщает серверу LES свои MAC- и ATM-адреса. Затем он регистрирует в LES все MAC-адреса узлов, которые он узнает при изучении своей локальной сети. Таким же образом поступают все пограничные ATM-LAN-коммутаторы, поэтому в сервере LES накапливается общая таблица соответствия MAC-адресов узлов локальных сетей ATM-адресам их пограничных коммутаторов.

Для взаимодействия с сервером LES каждый клиент LEC осуществляет прямое виртуальное соединение VCC с этим сервером, называемое Control Direct VCC. Это соединение устанавливается еще на стадии присоединения (Join) клиента LEC к эмулируемой сети. Под эмулируемой сетью понимают всю совокупность локальных сетей, взаимодействующих друг с другом через данный сервер LES и пограничные коммутаторы таким образом, как будто они работают в единой локальной сети Ethernet, объединенной обычными повторителями, мостами и коммутаторами.

Каждый ATM-LAN-коммутатор должен изначально знать только один адрес – ATM-адрес сервера адресов LES, чтобы установить с ним виртуальное соединение. При приходе кадра с неизвестным MAC-адресом пограничный ком-

мутатор может послать запрос серверу LES об ATM-адресе коммутатора, который обслуживает локальную сеть, где есть узел с данным MAC-адресом. Протокол передачи запроса на разрешение MAC-адреса и получения на него ответа является частью спецификации LANE и называется LE_ARP (LAN Emulation Address Resolution Protocol).

В сетях Ethernet часто применяют рассылку широковещательных сообщений, в сетях ATM эта функция не поддерживается для ограничения непроизводительной загрузки сети (хотя в сети ATM существует механизм многоадресной рассылки).

Для эмуляции широковещательных сообщений локальной сети Ethernet, в спецификации LANE определен сервер эмуляции, обеспечивающий рассылку широковещательных пакетов и пакетов с неизвестным адресом. Этот сервер называется BUS (Broadcast and Unknown Server). Он рассылает вышеуказанные пакеты во все пограничные коммутаторы, соединенные с локальными сетями и выполняющими функции LEC.

Сервер BUS имеет отдельный ATM-адрес, который сервер LES сообщает клиенту LEC при его присоединении к эмулируемой сети. Клиент LEC должен после этого установить с сервером BUS прямое виртуальное соединение Multicast Send VCC, по которому он будет пересылать кадры с широковещательными или неизвестными адресами. Сервер BUS добавляет каждого нового клиента LEC к мультивещательному соединению Multicast Forward VCC. Это соединение использует сервер BUS для одновременной (многоадресной) рассылки широковещательных кадров и кадров с неизвестными адресами всем пограничным коммутаторам эмулируемой сети.

Спецификация LANE рекомендует клиентам LEC делать LE_ARP-запрос серверу LES для кадра с неизвестным адресом и, не дожидаясь ответа, сразу же отправлять этот кадр через сервер BUS. Это ускоряет работу эмулируемой сети, так как кадры доходят до узла назначения широковещательным образом еще до того, как будет получен LE_ARP-ответ от сервера LES. После получения LE_ARP-ответа, LEC перестает посылать кадры для данного MAC-адреса широковещательно, а устанавливает виртуальное соединение Data Direct VCC с конкретным ATM-LAN-коммутатором (или же использует установленное ранее соединение с этим коммутатором) и передает остальные кадры с данным MAC-адресом уже по прямому каналу.

Эмуляция нескольких сетей

Обычно все пограничные коммутаторы образуют одну эмулируемую сеть. Спецификация LANE позволяет, если необходимо, образовать несколько эмулируемых сетей, не взаимодействующих непосредственно между собой: узлы, входящие в одну эмулируемую сеть, не получают кадры другой эмулируемой сети (какие бы типы MAC-адресов назначения не применялись: индивидуальные, групповые или широковещательные). Для этого в каждой из эмулируемых сетей необходимо активизировать отдельные серверы LES и BUS, а в погра-

ничных коммутаторах активизировать по одному элементу LEC для каждой эмулируемой сети (благодаря чему трафики локальных сетей Ethernet не смешиваются в сети ATM).

Для хранения информации о количестве активизированных эмулируемых сетей, а также об ATM-адресах серверов LES и BUS вводится еще один сервер – сервер конфигурации LECS (LAN Emulation Configuration Server). Этот сервер хранит список имен эмулируемых сетей, а также значения их основных параметров: ATM-адреса серверов LES и BUS каждой сети, тип сети (например, Ethernet или Token Ring), максимальный размер кадра, поддерживаемого этой сетью, и т. п. Поэтому каждый LEC по известному ему ATM-адресу при инициализации устанавливает соединение с единственным в сети сервером LECS и получает от LECS список всех эмулируемых сетей и их параметров.

На основании полученной информации LEC выбирает эмулируемую сеть, к которой он желает присоединиться и известить об этом LECS. Затем LEC выполняет процедуру присоединения к LES выбранной эмулируемой сети, регистрирует там MAC-адреса своих узлов и начинает работать с составной сетью. Протокол взаимодействия клиентской части протокола LANE, а именно LEC, с серверными частями этой спецификации LECS, LES и BUS называется LAN Emulation User-Network Interface (LUNI).

Система LANE обеспечивает эмуляцию сервиса локальных сетей типа Ethernet или Token Ring при передаче данных через сеть ATM и при этом не имеет значения, какой конкретно тип сетевого протокола применяется – для ATM решительно все равно, передается ли через ELAN трафик IP или IPX или AppleTalk или что-то еще. Иначе говоря, вся работа замкнута в рамках одной подсети без участия маршрутизаторов – ведь именно они должны разбираться в конкретных протоколах третьего уровня. При этом каждое оконечное ATM-устройство может поддерживать более одной ELAN-сети одновременно. Работающие в конкретной схеме сетевые протоколы не будут замечать, что между маршрутизаторами используется ATM.

Таким образом, система LANE позволяет организовывать различные подсети (т. е. участки, внутри которых не задействован алгоритм третьего уровня), на границах которых установлены маршрутизаторы, с помощью которых происходит коммутация пакетов от одной подсети к другой, и эти маршрутизаторы уже вынесены из структуры ATM.

Следовательно, для построения разветвленной IP-сети необходимо связать множество ELAN. Сеть ATM в коммутации между ELAN не используется. Сказанное иллюстрирует рис. 4.17, где показан пример построения большой сети на базе топологии ATM. На рис. 4.18 представлено видение этой же IP-сети с точки зрения маршрутизаторов.

На рис. 4.18 не показано, что между подсетями LAN-1 и LAN-4 существует гораздо более короткий путь, который проходит через узел ATM (см. рис. 4.17), не охваченный системой LANE, что является одним из недостатков использования маршрутизаторов.

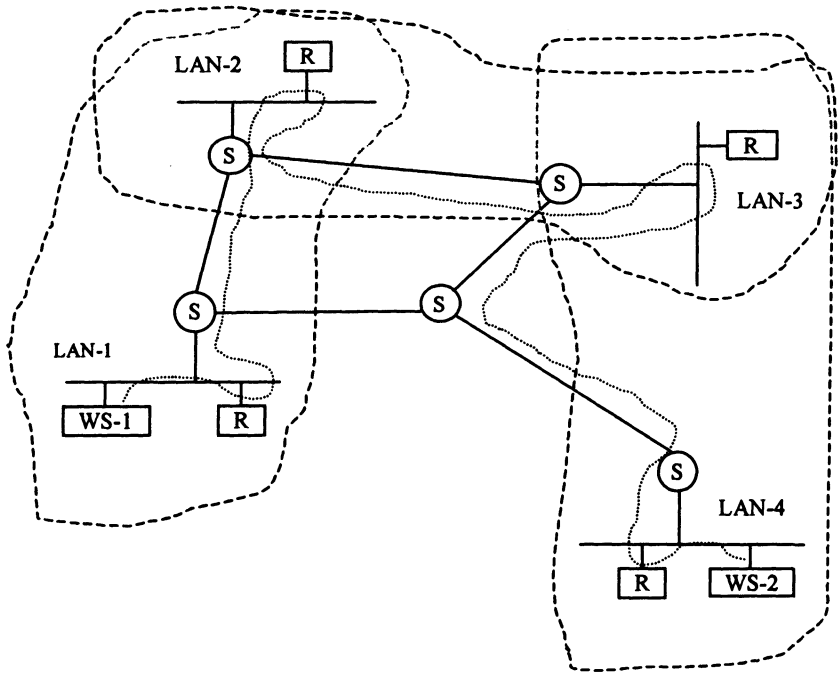


Рис. 4.17. Пример построения сети на базе технологии ATM:
S – ATM-коммутатор; R – маршрутизатор; WS – рабочая станция

Еще один их существенный недостаток состоит в том, что маршрутизаторы являются (наравне с каналами связи) узкими местами в сети, вносящими дополнительные задержки при передаче. Действительно, у каналов связи ограниченная пропускная способность, а у маршрутизаторов – ограниченная производительность, и при достаточно загруженной сети задержка коммутации будет весьма ощутимой (значительно большей, чем задержка передачи в случае использования ATM).

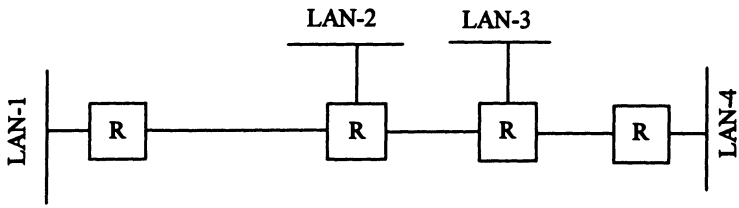


Рис. 4.18. Пример построения разветвленной IP-сети

Оптимизация процедуры выбора маршрутов через сеть ATM

Уменьшить задержку коммутации можно только одним путем – снизить количество транзитных маршрутизаторов либо увеличивая число сетей ELAN, повышая, таким образом, связность сети, либо попытаться обеспечить передачу пакетов от отправителя до получателя, минуя транзитные маршрутизаторы. Именно этот последний способ и реализован в системе MPOA. Суть способа состоит в оптимизации работы системы LANE, т. е. ускорении передачи данных между ELAN. Кроме оптимизации процедуры выбора маршрутов через сеть ATM система MPOA оптимизирует пути прохождения кадров данных в одной подсети, т. е. выполняет функцию моста. Таким образом, система LANE действует не только в рамках одной подсети, но и в рамках одного сегмента.

Система MPOA работает по архитектуре клиент–сервер, причем существовать она может только на базе LANE, т. е. это абсолютно неразрывные системы. Клиент и сервер MPOA расположены чаще всего в разных устройствах, подключенных к сети ATM, и соединены друг с другом через ELAN, т. е. находятся в одной подсети.

Главная задача клиента – выявить короткий путь до получателя и установить с ними ATM-соединение. Для обеспечения этого он выполняет коммутацию данных, но никогда не использует протоколы маршрутизации. На стороне отправителя (ingress client) выполняется обнаружение потока данных, которые должны быть переданы через LANE к маршрутизатору, где расположен сервер MPOA.

Когда поток данных устанавливается достаточно регулярным, то принимается решение об установлении прямого ATM-пути непосредственно до получателя. Начинается процесс установления ATM-соединения с получателем. Казалось бы чего проще? Но это только на первый взгляд. В самом деле, представим себе IP-сеть, некоторые узлы которой связаны через ATM-структуру (рис 4.19). Пусть от рабочей станции WS-1 исходит поток данных к станции WS-2. IP-адреса этих станций находятся в сетях, где нет устройств ATM, следовательно, между ними невозможно установить прямое соответствие. Однако в таблицах маршрутизации маршрутизаторов записано, что для достижения требуемой сети нужно переслать пакет по следующему пути: R1-R2-R3-R4. Маршрутизаторы R2, R3 и R4 объединены через LANE и поэтому пакеты будут проходить через транзитный маршрутизатор R3. При этом клиенты MPOA (MPC-1, MPC-2) пока еще ничего не знают друг о друге. Когда выяснится, что поток между станциями WS-1 и WS-2 достаточно активный, то начинается процедура выявления короткого пути (shortcut), задачей которой для MPC-1 является обнаружение MPC-2 и установления с ним прямого виртуального канала. Для выполнения этой задачи и необходима MPOA и все ее компоненты.

Если такой путь возможен, то клиент, инициировавший его установление, занесет сведения о получателе в свою внутреннюю таблицу (ingress cache), установит виртуальное соединение и в дальнейшем будет передавать все пакеты

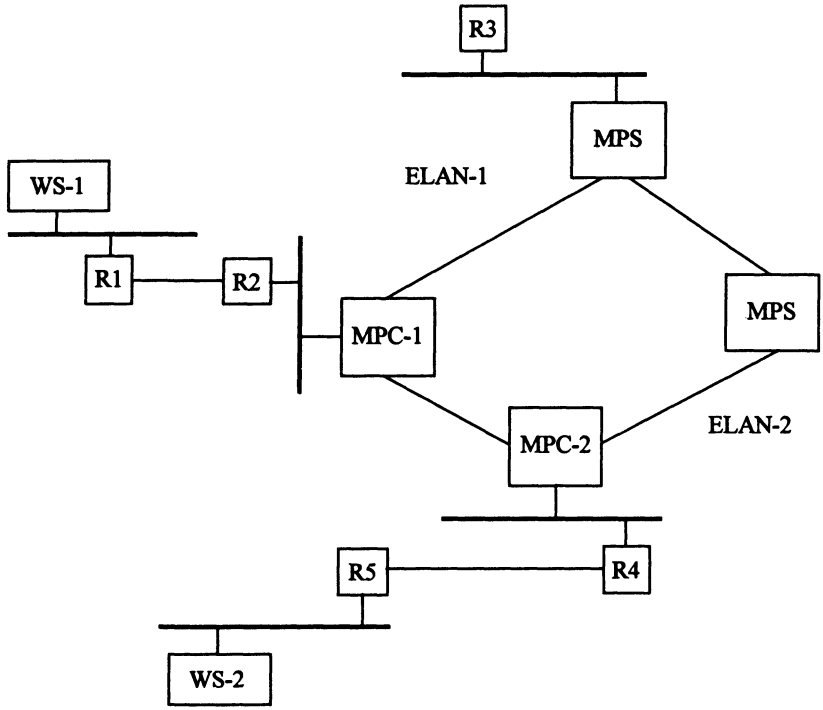


Рис. 4.19. Задача установления прямого пути между клиентами MPOA

в направлении MPC-2 по нему, минуя маршрутизатор R3. MPC-2 направляет все пакеты, принятые от MPC-1 в свою IP-сеть. MPC-1 и MPC-2 расположены в разных сетях, в разных ELAN и могут быть организованы по-разному. В частности, ELAN-1 может работать по протоколам Ethernet, а ELAN-2 – по протоколам Token Ring. Следовательно, MPC-2 должен произвести еще и преобразование заголовков кадров второго уровня для перехода между этими двумя протоколами. Необходимость такого перехода передается в MPC-2 от сервера MPS, и параметры этого перехода заносятся в его внутреннюю таблицу (egress cache).

MPOA-сервер – MPS перенаправляет информацию сетевого уровня между клиентами. Для этих целей, используется Next Hop Resolution Protocol (NHRP), реализуемый в Next Hop Servers (NHS). Дело в том, что на пути между клиентами может быть несколько MPS, т. е. эти серверы объединены в сеть, и необходимо найти путь в этой сети между сервером–отправителем и сервером–получателем. Кроме того, с помощью MPS определяются параметры необходимых преобразований в клиентах на кадровом уровне модели. Таким образом, серверы служат только лишь для организации правильного взаимодействия между клиентами и помогают им обнаружить друг друга.

Спецификация MPOA описывает несколько типов MPOA-устройств.

- Пограничное устройство MPOA (MPOA edge device). Оно состоит из клиента MPOA, клиента LEC и порта связи с местной IP-сетью. Иначе говоря, это устройство позволяет транслировать IP-пакеты из местной подсети в другую подсеть через MPOA.
- MPOA Host. Это устройство отличается от предыдущего тем, что оно не имеет связи с местной IP подсетью, т. е. это просто рабочая станция, оснащенная стеком ATM и имеющая IP адрес.
- Маршрутизатор. Он состоит из клиента LEC, сервера MPOA (MPS), который в свою очередь включает еще и Next Hop Server (NHS).

Как видим, во всех устройствах присутствует клиент LEC, причем спецификация допускает подключение MPC и MPS сразу к нескольким клиентам LEC, что дает возможность подвести к одному устройству несколько подсетей. Однако, у одного клиента LEC не может быть более одного MPOA-устройства.

Для взаимодействия всех устройств спецификацией MPOA предусмотрен целый набор служебных потоков, которые нужны, во-первых, для распознавания друг друга, а, во-вторых, для обмена служебной информацией в процессе работы. Так, MPC и MPS обнаруживают друг друга с помощью сервера LECS, т. е. при настройке LANE. Значит, в эту настройку должны быть включены параметры MPOA, конкретно, имена устройств, их типы и ATM-адреса.

Управляющие потоки между клиентом и сервером используются для управления внутренними таблицами – Cache Management. Процедуры запросов и ответов позволят клиенту-отправителю (Ingress MPC) устанавливать прямое соединение, а клиенту-получателю (Egress MPC) – получить необходимые параметры для преобразований кадров перед выдачей их в свою подсеть. Кроме того, с помощью служебных потоков клиент-получатель или сервер в случае, если он обнаружил, что содержимое внутренней таблицы перестало соответствовать реальной ситуации на сети, может послать очищающее сообщение – *Purge message*.

Служебные потоки между серверами несут в себе информацию для работы процедуры маршрутизации и протокола NHRP (Next Hop Resolution Protocol), который определяет пути между серверами через сеть ATM. Поток данных между клиентами служит, главным образом, для передачи данных по ранее установленному прямому пути, минуя маршрутизатор.

4.5. Технология мобильных сетей

Принципы построения цифровых сетей сотовой подвижной связи

Сотовые технологии обеспечивают связь между подвижными абонентами (ячейками) и стационарными серверами по радиоканалу. Поэтому сотовую связь и называют мобильной. Основой развития мобильных сетей являются сотовые топологии. Доступ к радиоканалу осуществляется одним из следующих способов:

• случайный доступ (метод ALOHA). Применяют только при малых нагрузках. Его развитием стал метод МДКН/ОС, используемый в локальных сетях;

• технология CDMA. За каждым абонентом закрепляют фиксированную частоту, на которой с помощью временного мультиплексирования выделяется фиксированный временный слот (здесь подробно не рассматривается).

• технология TDMA (Time Division Multiple Access). Временно́е мультиплексирование с выделением слота по требованию. Требования отсылают в короткие интервалы времени (слоты запросов), при коллизиях запросы повторяют. Базовая станция выделяет свободные информационные слоты, сообщая их источнику и получателю.

К настоящему времени разработано три основных стандарта перспективных цифровых сетей сотовой подвижной связи (ССПС) с макросотовой топологией сетей и радиусом соты, соответствующим максимальной дальности связи в радиальных системах (около 35 км): общеевропейский стандарт GSM; американский стандарт ADC (D-AMPS); японский стандарт JDC. Хотя эти стандарты на цифровые ССПС и отличаются своими характеристиками, они построены на единых принципах и концепциях, использованных в стандарте GSM, и отвечают требованиям современных информационных технологий (табл. 4.4).

Таблица 4.4. **Характеристики стандартов ССПС**

Характеристика стандарта	Стандарт ССПС		
	GSM	ADC	JDC
Метод доступа	TDMA	TDMA	TDMA
Разнос частот, кГц	200	30	25
Количество речевых каналов на несущую	8	3	3
Скорость преобразования речи, кбит/с	13	8	11,2
Алгоритм преобразования речи	RPE-LTP	VSELP	VSELP
Общая скорость передачи, кбит/с	270	48	42
Эквивалентная полоса на речевой канал, кГц	25	10	8,3
Вид модуляции	0,3 GMSK	$\pi/4$ DQPSK	$\pi/4$ DQPSK
Требуемое отношение несущая/интерференция (С/I), дБ	9	16	13

Окончание табл. 4.4

Характеристика стандарта	Стандарт ССПС		
	GSM	ADC	JDC
Рабочий диапазон частот, МГц	935...965 890...915	824...840 869...894	810...826 940...956 1429...1441 1447...1489 1453...1465 1501...1513
Радиус соты, км	0,5...35	0,5...20	0,5...20

Американский стандарт ADC (D-AMPS) разрабатывался для отличных от Европы условий: диапазон частот 800 МГц и работа в общей с существующей аналоговой ССПС AMPS полосе частот. В этом случае для цифровой ССЦС необходимо было сохранить частотный разнос каналов 30 кГц, используемый в AMPS, и обеспечить одновременную работу абонентских радиостанций как в аналоговом, так и в цифровом режимах. Применение специально разработанного речевого кодека (VSELP), имеющего скорость преобразования речевого сигнала 8 кбит/с, и цифровой дифференциальной квадратурной фазовой манипуляции со сдвигом $\pi/4$ позволило в режиме TDMA организовать три речевых канала на одну несущую с разносом канальных частот 30 кГц.

Японский стандарт JDC во многом совпадает с американским. Основное отличие состоит в использовании другого частотного диапазона, дуплексного разноса полос частот приема и передачи – 55 МГц при разносе каналов 25 кГц. Стандарт JDC адаптирован также к диапазону 1500 МГц.

Все рассмотренные стандарты обеспечивают взаимодействие цифровых ССПС с ISDN и PDN и гарантируют высокое качество передаваемых сообщений в режимах открытой или закрытой (засекреченной) передачи.

Структура уровней в модели OSI применительно к стандарту GSM показана на рис. 4.20.

Принципы построения цифровых ССПС позволили использовать при организации сотовых сетей новые более эффективные модели повторного использования частот, чем в аналоговых сетях. В результате, без увеличения общей полосы частот системы связи, значительно возросло число каналов на одну соту (ячейку). В первую очередь, это относится к стандарту GSM. Вид модуляции, способы кодирования и формирования сигналов в каналах связи, принятые в GSM, обеспечивают прием сигналов с отношением сигнал/интерференция C/I – 9 дБ, в то время как в аналоговых системах тот же показатель равен 18 дБ. Поэтому передатчики базовых станций (BTS), работающие на совпадающих частотах, могут находиться на более близких расстояниях без потери высокого качества приема сообщений.

Рис. 4.20. Связь уровней GSM и OSI

Первыми моделями повторного использования частот, которые применялись в аналоговых ССПС, были модели с круговыми диаграммами направленности (ДН) антенн BTS. В сетях цифровых ССПС для сот с круговой ДН антенн применяют модель повторного использования частот, включающую 7 или 9 BTS. На рис. 4.21. представлена модель повторного использования частот для семи BTS. Модель с круговой ДН антенн предполагает передачу сигнала одинаковой мощности по всем направлениям, что для абонентских станций эквивалентно приему помех со всех направлений.

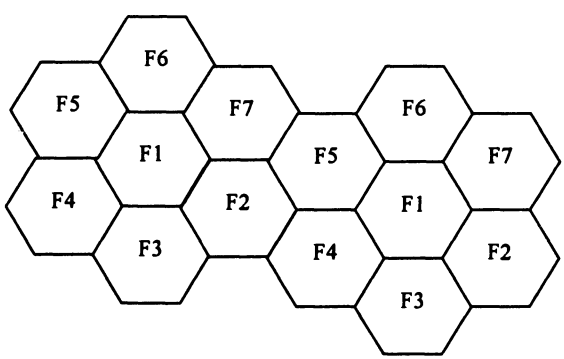


Рис. 4.21. Модель повторного использования частот для семи BTS

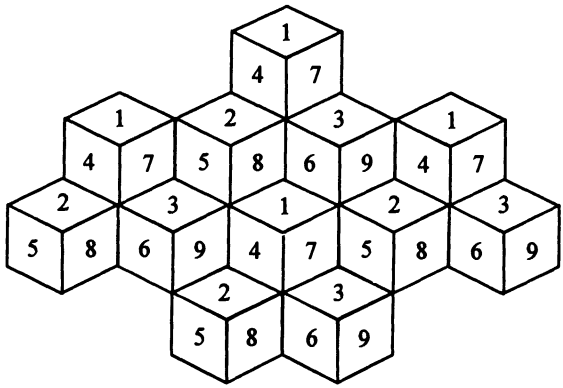


Рис. 4.22. Модель повторного использования частот с трехсекторными сотами

Эффективным способом снижения уровня помех является использование секторных антенн. В секторе направленной антенны сигнал излучается в одну сторону, а уровень в противоположном направлении сокращается до минимума. Разбиение сот на секторы позволяет более часто повторно применять частоты в сотах. Общеизвестная модель повторного использования частот в разбитых на секторы сотах включает три соты и три BTS. В таком случае задействуют три 120-градусные антенны BTS с формированием девяти групп частот (рис. 4.22).

Самую высокую эффективность использования полосы частот, т. е. наибольшее число абонентов сети в выделенной полосе частот, обеспечивает разработанная фирмой Motorola (США) модель повторного использования частот, включающая две BTS. Как следует из схемы, изображенной на рис. 4.23,

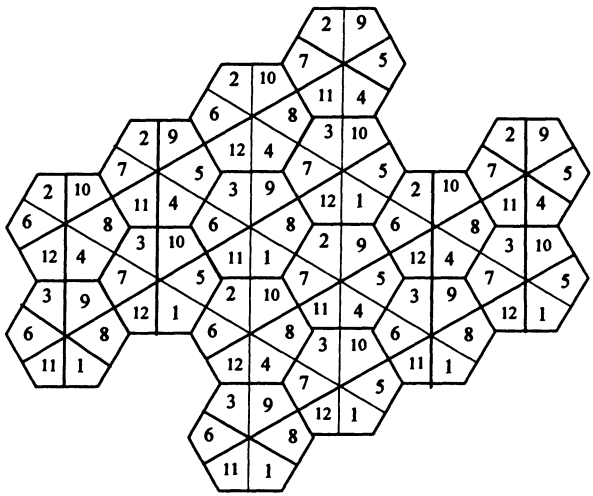


Рис. 4.23. Модель повторного использования частот в двух соседних сотах

каждую частоту используют дважды в пределах модели, состоящей из четырех BTS. Благодаря этому каждая из четырех BTS в пределах действия шести 60-градусных антенн может работать на 12 группах частот.

Например, в сети GSM с общей полосой 7,2 МГц (36 частот) модель повторного использования частот двумя BTS позволяет на одной BTS одновременно применять 18 частот (в модели с тремя BTS таких частот 12).

В любой ССПС емкость сетей зависит от числа каналов связи в соте:

$$N = \frac{1}{k} \cdot \frac{F}{f}, \tag{4.3}$$

где F – полоса частот ССПС; $f = F_k / n$ – эквивалентная полоса частот, приходящаяся на один речевой канал (F_k – полоса канала связи; n – число временных позиций в TDMA-кадре); F / f – число каналов связи; k – коэффициент повторного использования частот.

В соответствии с определениями ITU-T (International Telecommunication Union – Telecommunications Standardization Sector), сеть GSM предоставляет следующие виды услуг:

- перенос информации (bearer services);
- предоставление связи (teleservices);
- дополнительные услуги (supplementary services).

Кроме того, предоставляются разнообразные услуги передачи данных. Абоненты GSM могут осуществлять обмен информацией с абонентами ISDN, обычных телефонных сетей, сетей с коммутацией пакетов и сетей связи с коммутацией каналов, используя различные методы и протоколы доступа, например X.25. Возможна передача факсимильных сообщений, реализуемых при наличии соответствующего адаптера для факс-аппарата. Уникальной возможностью GSM, которой не было в старых аналоговых системах, является двунаправленная передача коротких сообщений SMS (Short Message Service) до 160 байт, передаваемых в режиме с промежуточным хранением данных. Адресату, являющемуся абонентом SMS, может быть послано сообщение, после которого отправителю посылается подтверждение о получении. Короткие сообщения можно использовать в режиме широковещания, например, для того, чтобы извещать абонентов об изменении условий дорожного движения в регионе.

Структурная схема и состав оборудования мобильных сетей связи

Функциональное построение и интерфейсы, принятые в стандарте GSM, иллюстрирует структурная схема, представленная на рис. 4.24. В схему входит центр коммутации подвижной связи MSC (Mobile Switching Centre), оборудование базовой станции BSS (Base Station System), центр управления и обслуживания OMC (Operations and Maintenance Centre) и подвижные станции MS (Mobile Stations).

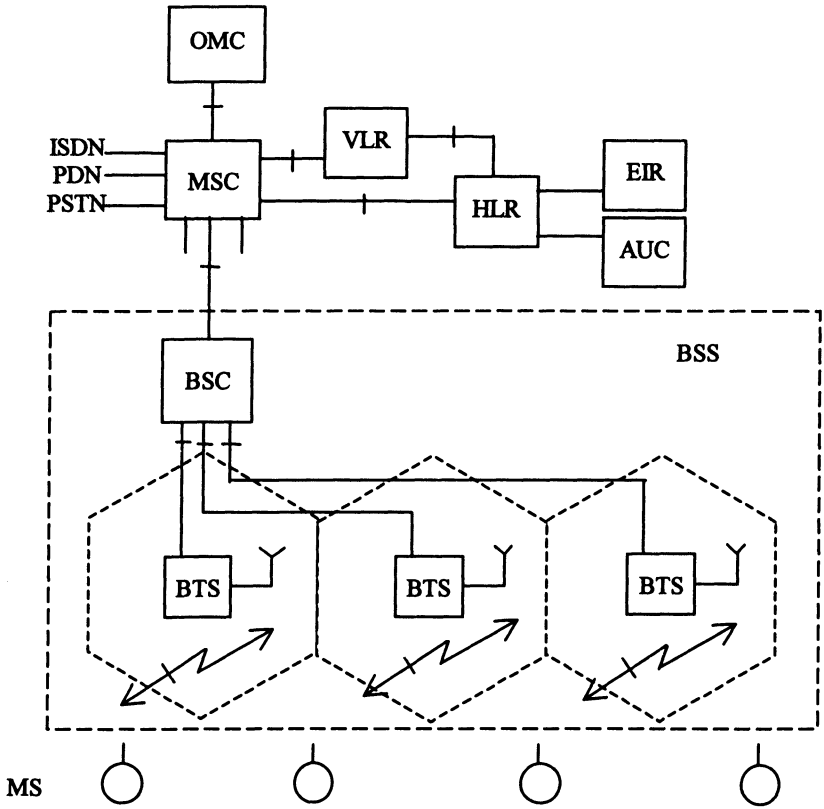


Рис. 4.24. Структурная схема мобильной сети

Функциональное сопряжение элементов системы осуществляется рядом интерфейсов. Все сетевые функциональные компоненты в стандарте GSM взаимодействуют в соответствии с системой сигнализации МККТТ N7 (CCITT SS. N7).

Центр коммутации подвижной связи обслуживает группу сот и обеспечивает все виды соединений, необходимых для работы подвижной станции. MSC аналогичен ISDN коммутационной станции и представляет собой интерфейс между фиксированными сетями (PSTN, PDN, ISDN и т. д.) и сетью подвижной связи. Он обеспечивает маршрутизацию вызовов и функции управления вызовами. Кроме выполнения функций обычной ISDN коммутационной станции, на MSC возложены функции коммутации радиоканалов, к которым относится «эстафетная передача». В процессе этой передачи достигается непрерывность связи при перемещении подвижной станции из зоны в зону и переключение рабочих каналов в соте при появлении помех или неисправностях.

Центр коммутации осуществляет постоянное слежение за подвижными станциями, используя регистры положения (HLR) и перемещения (VLR). В HLR хранится та часть информации о местоположении какой-либо подвижной станции, которая позволяет центру коммутации доставить вызов станции. Регистр HLR содержит международный идентификационный номер подвижного абонента (IMSI). Его используют для опознавания подвижной станции в центре аутентификации (AUC).

Регистр перемещения VLR – обеспечивает контроль за передвижением подвижной станции из зоны в зону. Он обеспечивает функционирование подвижной станции за пределами зоны, контролируемой HLR. Когда в процессе перемещения подвижная станция переходит из зоны действия одного контроллера базовой станции BSC, объединяющего группу базовых станций, в зону действия другого BSC, ее регистрирует новый BSC, и в VLR заносится информация о номере области связи, которая обеспечит доставку вызовов, подвижной станции. Для сохранности данных, находящихся в HLR и VLR, в случае сбоев предусмотрена защита устройств памяти этих регистров.

Для несанкционированного использования ресурсов системы связи предусмотрены механизмы аутентификации – удостоверения подлинности абонента. Центр аутентификации состоит из нескольких блоков и формирует ключи и алгоритмы аутентификации. С его помощью проверяются полномочия абонента и осуществляется его доступ к сети связи. AUC принимает решения о параметрах процесса аутентификации и определяет ключи шифрования абонентских станций на основе базы данных, сосредоточенной в регистре идентификации оборудования (EIR – Equipment Identification Register).

Каждый подвижный абонент на время пользования системой связи получает стандартный модуль подлинности абонента (SIM), который содержит: международный идентификационный номер (IMSI), свой индивидуальный ключ аутентификации (Ki), алгоритм аутентификации (A3).

С помощью заложенной в SIM информации, в результате взаимного обмена данными между подвижной станцией и сетью, осуществляется полный цикл аутентификации и разрешается доступ абонента к сети.

Оборудование базовой станции состоит из контроллера базовой станции BSC или BTS. Контроллер базовой станции может управлять несколькими приемопередающими блоками. Он управляет распределением радиоканалов, контролирует соединения, регулирует их очередность, обеспечивает режим работы с прыгающей частотой, модуляцию и демодуляцию сигналов, кодирование и декодирование сообщений, кодирование речи, адаптацию скорости передачи для речи, данных и вызова, определяет очередность передачи сообщений персонального вызова.

В рамках стандарта GSM приняты 5 классов подвижных станций, различающиеся по мощности от 20 (1 класс) до 0,8 Вт (5 класс).

Подвижный абонент и станция независимы друг от друга. Как уже отмечалось, каждый абонент имеет свой международный идентификационный номер

(IMSI), записанный на его интеллектуальную карточку. Такой подход позволяет устанавливать радиотелефоны, например, в такси и автомобилях, сдаваемых на прокат. Каждой подвижной станции также присваивается свой международный идентификационный номер (IMEI). Этот номер используется для предотвращения доступа к сетям GSM похищенной станции или станции без полномочий.

Структура TDMA-кадров и формирование сигналов в стандарте GSM

Стандарт TDMA широко применяют в современных цифровых системах подвижной связи. В отличие от систем частотного разделения, все абоненты системы TDMA работают в одном и том же диапазоне частот, но при этом каждый имеет временные ограничения доступа. Каждому абоненту выделен временной промежуток (кадр), в течение которого ему разрешено «вещание». Когда один абонент завершает вещание, разрешение передается другому, затем третьему и т. д. После того, как обслужены все абоненты, процесс начинается сначала. С точки зрения абонента его активность носит пульсирующий характер. Чем больше абонентов, тем реже каждому из них предоставляется возможность передать свои данные, тем, соответственно, меньше данных он сможет передать.

В результате анализа различных вариантов построения цифровых ССПС в стандарте GSM принята комбинация методов множественного доступа TDMA и FDMA. Общая структура временных кадров GSM показана на рис. 4.25.

Длина периода последовательности в этой структуре, которая называется гиперкадром, равна $T_г$ = 3 ч 28 м 53 с 760 мс (12533,76 с). Гиперкадр делится на 2048 суперкадров, каждый из которых имеет длительность $T_с$ = 12533,76 / 2048 = = 6,12 с. Суперкадр состоит из мультикадров. Для организации различных каналов связи и управления в стандарте GSM используют два типа мультикадров:

- 26-позиционные TDMA-кадры мультикадра;
- 51-позиционные TDMA-кадры мультикадра.

В суперкадре может быть 51 мультикадр первого типа или 26 мультикадров второго типа. Длительности мультикадров соответственно равны:

$$T_м = 6120 / 51 = 120 \text{ мс};$$

$$T_м = 6120 / 26 = 235{,}385 \text{ мс } (3060 / 13 \text{ мс}).$$

Длительность каждого TDMA-кадра

$$T_к = 120 / 26 = 235{,}385 / 51 = 4{,}615 \text{ мс } (60 / 13 \text{ мс}).$$

В периоде последовательности каждый TDMA-кадр имеет свой порядковый номер (NF) от 0 до NF_{max}, где

$$NF_{max} = (26 \times 51 \times 2048) - 1 = 2715647.$$

Таким образом, гиперкадр состоит из 2715647 TDMA-кадров. Необходимость такого большого периода гиперкадра объясняется требованиями применяемого процесса криптографической защиты, в котором номер кадра NF используется как входной параметр.

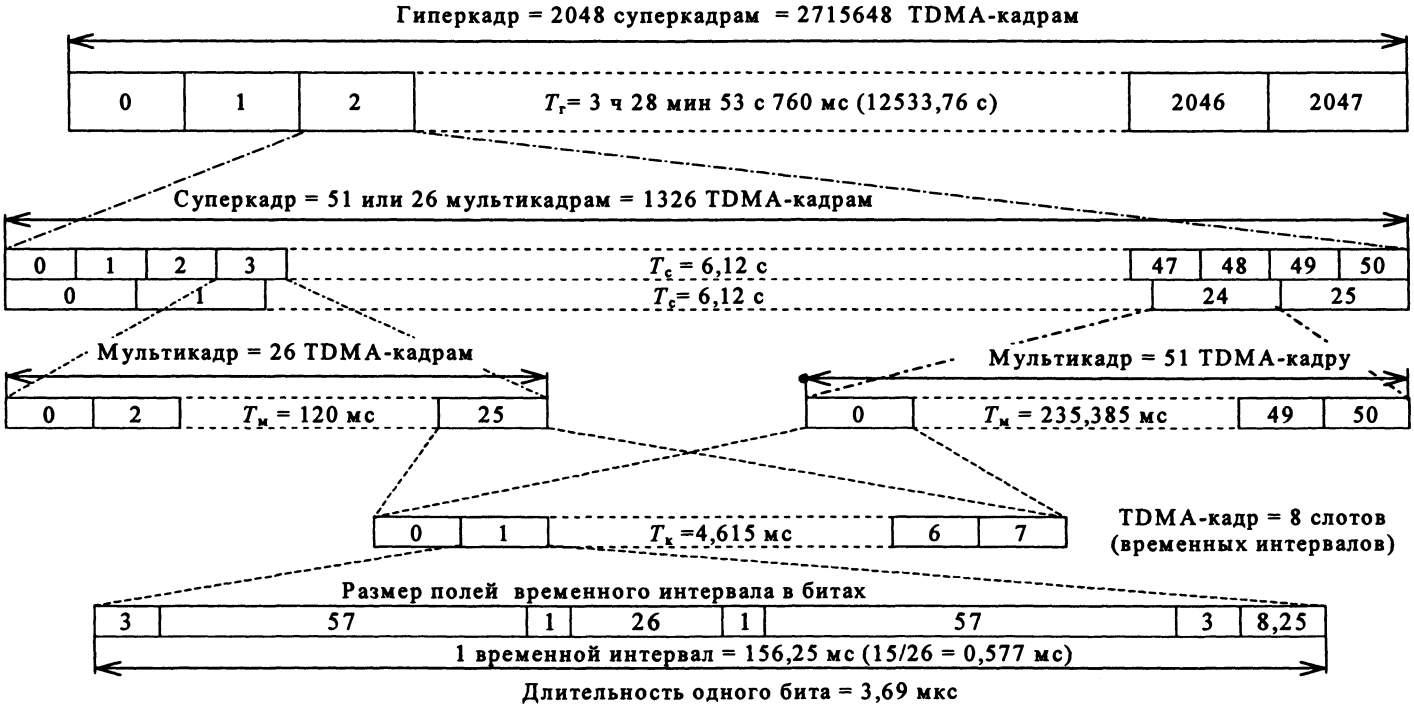


Рис. 4.25. Структура временных кадров GSM

TDMA-кадр делится на восемь временны́х интервалов (слотов) с периодом

$$T_0 = 60 / 13{:}8 = 576{,}9 \text{ мкс } (15 / 26 \text{ мс}).$$

Мультикадр GSM продолжительностью 120 мс, разделен на 26 или 51 TDMA-кадра (фрейма), каждый из которых состоит из восьми выделенных временны́х интервалов – слотов. В мультикадре 24 фрейма содержат пользовательскую информацию, передающуюся по логическим каналам передачи сообщений. Оставшиеся два фрейма, в середине и в конце мультикадра, система использует для передачи управляющей информации по так называемым ассоциированным управляющим каналам. В ходе каждого звонка система выделяет один из этих каналов, обеспечивающий в системе возможность передачи сигнальной информации вне пределов основной полосы передачи информации – возможность, не существовавшая в системах первого поколения ССПС. Два пакета по 57 бит данных каждого временного слота предназначены для передачи пользовательской информации, а один разделительный бит в каждом пакете является флагом для того, чтобы отделить передачу речи от других передач. В состав слота включены также 26 бит эквалайзерной (синхронизирующей) последовательности. Слот начинается и завершается концевиком, состоящим из 3 бит логических нулей. Межслотовые интервалы, состоящие из 8,25 бит, предохраняют от перекрытия на базовой станции сигналов, поступающих с разных мобильных терминалов. При передаче 156,25 бит за 577 мкс, скорость передачи равна 270,833 кбит/с, битовый интервал – 3,69 мкс. Систему *GSM* создавали исходя из предположения, что приемники обеспечивают точный прием, если множественные пути сигналов имеют разницу в задержке до 16 мкс, что составляет более четырех битовых интервалов.

В GSM используется гауссовская частотная манипуляция с минимальным сдвигом (GMSK) и индексом манипуляции 0,3, при которой применяемая частота среза равна 3 дБ при частоте 81,25 кГц (0,3 битовой скорости). Эффективность модуляции сигналов при скорости передачи порядка 271 кбит/с, при разделении каналов в 200 кГц, равна 1,35 бит/с/Гц. При кодировании скорость источника составляет 13 кбит/с, а скорость передачи, включая распознавание ошибок и корректирующие коды, – 22,8 кбит/с.

В стандарте GSM использованы комбинированная TDMA/FDMA-схема организации каналов и принцип медленных скачков по частоте при передаче сообщений во временных кадрах.

Принятая структура TDMA кадров и принципы формирования сигналов в стандарте GSM в совокупности с методами канального кодирования позволили снизить требуемое для приема отношение сигнал/помеха до 9 дБ, тогда как в стандартах аналоговых сотовых сетей связи оно составляет 18 дБ.

Организация физических и логических каналов в стандарте GSM

Физический канал в стандарте GSM представляет собой комбинацию временного и частотного разделения сигналов и определяется как последователь-

ность радиочастотных каналов (с возможностью перескоков по частотам) и временных окон TDMA-кадра.

Стандарт GSM разработан для создания ССПС в следующих полосах частот: 890...915 МГц – для передачи подвижными станциями (линия «вверх»); 935...960 МГц – для передачи базовыми станциями (линия «вниз»). Частотные планы ССПС, включая стандарт GSM, показаны на рис.4.26.

Каждая из полос, выделенных для GSM, разделена на частотные каналы. Разнос каналов составляет 200 кГц, что позволяет организовать в GSM 124 частотных канала, которые распределяются в соответствии с размещением сот. Частоты, выделенные для передачи от подвижной станции на базовую и в обратном направлении, группируют парами, организуя дуплексный канал с разносом 45 МГц. Эти пары частот сохраняются и при перескоках частоты. Каждая ячейка (сота) характеризуется присвоением определенного количества пар частот от 1 до 15 (не более).

Каждая частотная несущая содержит 8 физических каналов, размещенных в 8 временны́х интервалах (слотах) в пределах TDMA-кадра. Каждый физический канал использует один и тот же слот в каждом временно́м TDMA-кадре.

До формирования физического канала сообщения и данные, представленные в цифровой форме, группируются и объединяются в логические каналы двух типов: каналы связи для передачи кодированной речи или данных (TCH), каналы управления для передачи сигналов управления и синхронизации. Различают 4 вида каналов управления: BCCH (Broadcast Control CHannels) – каналы передачи сигналов управления, CCCH (Common Control CHannels) – общие каналы управления, DCCH (stand-alone Dedicated Control CHannels) – индивидуальные каналы управления, ACCH (Associated Control CHannels) – совмещенные каналы управления.

Каналы передачи сигналов управления (BCCH) используют только в направлении с базовой станции на все подвижные станции. Они несут информацию, необходимую подвижным станциям для работы в системе. Различают три вида каналов передачи сигналов управления :

FCCH (Frequency Correction CHannel) – канал подстройки частоты, используемый для синхронизации несущей в подвижной станции. По этому каналу передают немодулированную несущую с фиксированным частотным сдвигом относительно номинального значения частоты канала связи.

SCH (Synchronisation CHannel) – канал синхронизации, по которому передают информацию на подвижную станцию с кадровой (временно́й) синхронизацией.

BCCH (Broadcast Control CHannel) – канал управления передачей, обеспечивающей передачу основных команд по управлению передачей (номер общих каналов управления тех из них, которые объединены с другими каналами, в том числе и с физическими и т. д.)

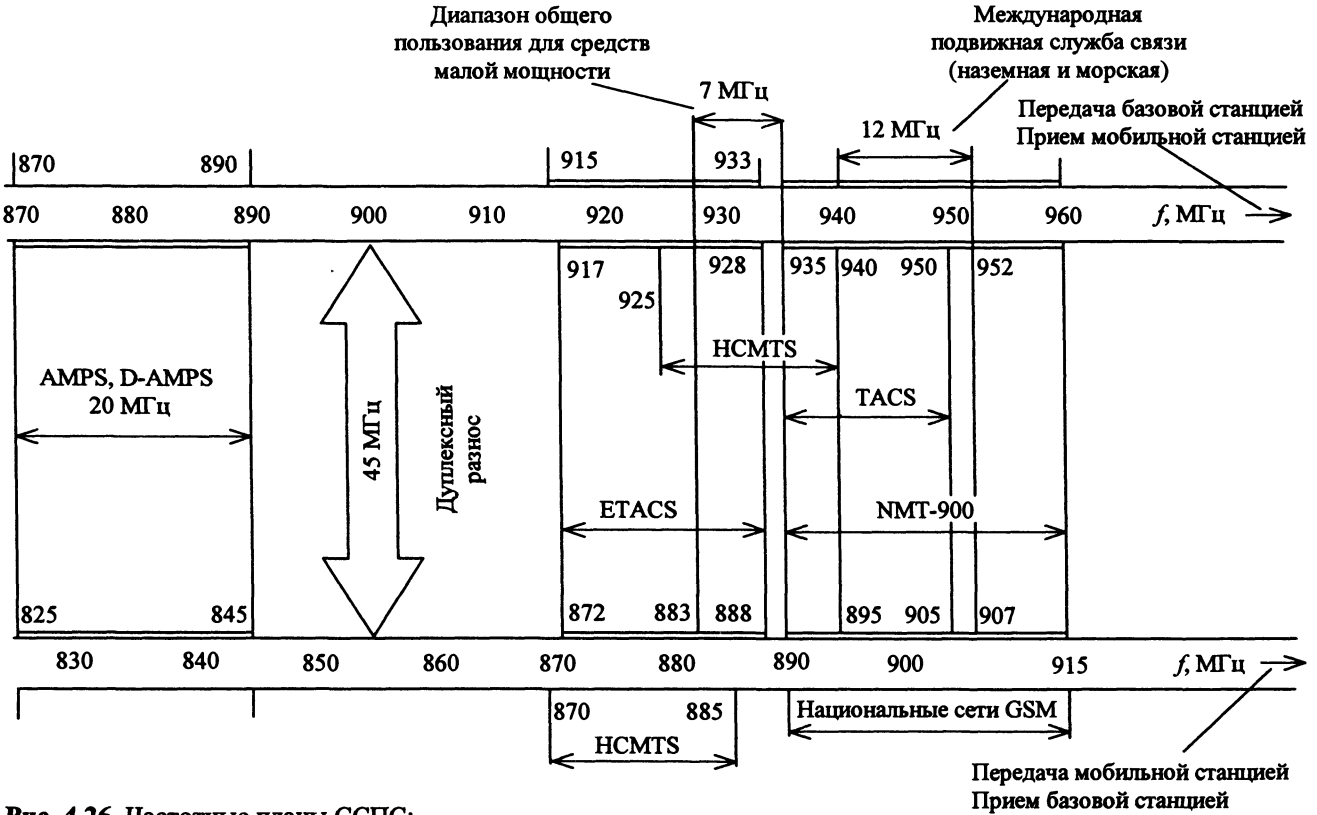


Рис. 4.26. Частотные планы ССПС:
GSM – общеевропейский стандарт на цифровые ССПС; AMPS – стандарт на аналоговые ССПС Северной Америки; D – AMPS (ADS) – стандарт на цифровые ССПС Северной Америки; TACS (ETACS) – стандарт Великобритании на аналоговую ССПС; NMT-900 – стандарт скандинавских стран на аналоговую ССПС; HCMTS – стандарт на аналоговые ССПС Японии

Отображение логических каналов на физические каналы осуществляется через процессы кодирования и шифрования передаваемых сообщений. Для защиты логических каналов от ошибок, имеющих место в процессе передачи, используют три вида кодирования:

блочное – для быстрого обнаружения ошибок при приеме;

сверточное – для исправления одиночных ошибок;

перемежение – для преобразования пакетов ошибок в одиночные.

Для защиты каналов от подслушивания в каналах связи и управления применяют шифрование.

Аспекты безопасности в мобильных сетях

Сотовые системы подвижной связи нового поколения могут принять всех потенциальных пользователей, если будет гарантирована безопасность связи: секретность и аутентификация. Секретность должна исключить возможность извлечения информации из каналов связи кому-либо, кроме санкционированного получателя. Проблема аутентификации заключается в том, чтобы помешать кому-либо, кроме санкционированного пользователя (отправителя), подменить канал, т. е. получатель должен быть уверен, что в настоящий момент он принимает сообщение от санкционированного пользователя. Основным способом обеспечения секретности является шифрование. Новой концепцией использования шифрования является аутентификация сообщений.

Аутентификацию сообщений через шифрование осуществляют за счет включения в текст так называемого кода идентификации (т. е. фиксированного или зависящего от передаваемых данных слова, которое знают отправитель и получатель, или которое они могут выделить в процессе передачи). Получатель расшифровывает сообщение и путем сравнения получает удостоверение, что принимаемые данные являются именно данными санкционированного отправителя.

К системе шифрования предъявляют следующие основные требования:

- обеспечение нелинейной связи между исходным текстом и зашифрованным;
- возможность изменения параметров шифрования во времени.

Если алгоритмы шифрования отвечают первому требованию, то, не зная ключа, исключена возможность изменить код идентификации. Второе требование исключает возможность нарушения работы системы за счет воспроизведения противником перехваченного ранее сообщения.

Одним путем из обеспечения этих требований является применение синхронных систем передачи, но при этом необходимы системы цикловой и тактовой синхронизации, что во многих случаях неприемлемо. Второй путь состоит во включении в информационную последовательность (каждое сообщение) временных меток, так чтобы зашифрованные данные были бы однозначно с ними связаны.

Алгоритмы шифрования делят на два класса:
классические,
с открытым ключом.

Классические алгоритмы используют один ключ для шифрования и дешифрования. Алгоритмы с открытым ключом используют два ключа :

1) для перехода от нешифрованного текста к шифрованному;
2) для обратного перехода от шифрованного к нешифрованному.

Алгоритм шифрования с открытым ключом RSA обеспечивает высокую степень безопасности передачи речевых сообщений.

В стандарте GSM термин «безопасность» подразумевает исключение несанкционированного использования системы и обеспечение секретности переговоров подвижных абонентов. В стандарте GSM определены следующие механизмы безопасности:

- аутентификация;
- секретность передачи данных;
- секретность абонента;
- секретность направлений соединения абонентов.

Защита сигналов управления и данных пользователя осуществляется только по радиоканалу.

Рассмотрим механизмы аутентификации и секретности передачи данных.

Механизмы аутентификации. Для исключения несанкционированного использования ресурсов системы связи определены механизмы аутентификации – удостоверения подлинности абонента.

Каждый подвижный абонент на время пользования системой связи получает стандартный модуль подлинности абонента (SIM), который содержит:

- международный идентификационный номер подвижного абонента (IMSI);
- свой индивидуальный ключ аутентификации (Ki);
- алгоритм аутентификации (A3).

С помощью заложенной в SIM информации в результате взаимного обмена данными между подвижной станцией и сетью осуществляется полный цикл аутентификации и разрешается доступ абонента к сети.

Процедура проверки сетью подлинности абонента изображена на рис. 4.27.

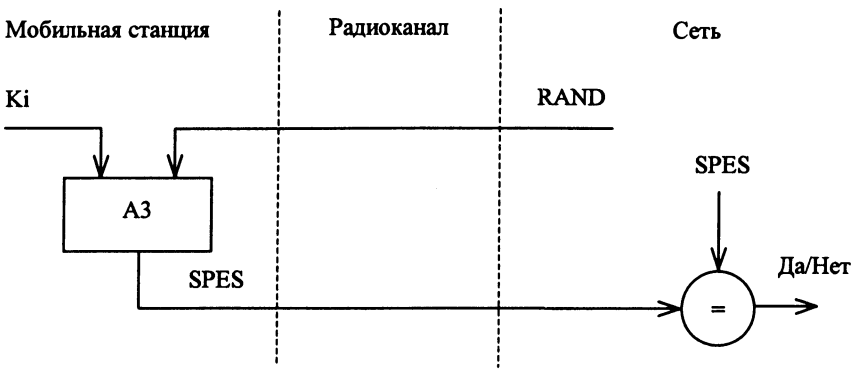


Рис. 4.27. Процедура аутентификации абонента

Процедура проверки сетью подлинности абонента реализуется следующим образом. Сеть передает случайный номер (RAND) на подвижную станцию. На ней с помощью ключа Ki и алгоритма аутентификации A3 определяется значение отклика (SRES), т. е.

$$SRES = Ki[RAND].$$

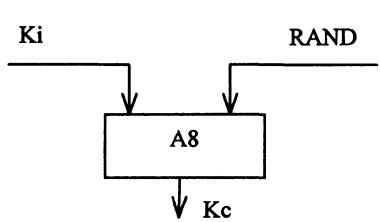


Рис. 4.28. Схема получения ключа шифрования

Подвижная станция посылает вычисленное значение SRES в сеть, которая сверяет значение принятого SRES со значением SRES, вычисленным сетью. Если оба значения совпадают, подвижная станция приступает к передаче сообщений. В противном случае связь прерывается, и индикатор подвижной станции показывает, что опознавание не состоялось. Для обеспечения секретности вычисление SRES происходит в рамках SIM.

Секретность передачи данных. Для обеспечения секретности передаваемой по радиоканалу информации введен следующий механизм защиты. Все конфиденциальные сообщения необходимо передавать в режиме защиты информации. Алгоритм формирования ключей шифрования (A8) хранится в модуле SIM. После приема случайного номера RAND подвижная станция вычисляет, кроме отклика SRES, также и ключ шифрования (Kc), используя RAND, Ki и алгоритм A8 (рис. 4.28.)

$$Kc = Ki\,[RAND].$$

Ключ шифрования Kc не передается по радиоканалу. Как подвижная станция, так и сеть вычисляют ключ шифрования, который используют другие подвижные абоненты. По причине секретности вычисление Kc происходит в SIM.

Спутниковые системы подвижной связи

Примером спутниковой системы подвижной связи является система «Глобалстар».

Эта система, используя сеть низкоорбитальных спутников или объектов (НОО), предоставляет услуги по передаче голоса, данных, обмену сообщениями, факсимиле и услуги определения местонахождения для клиентов во всем мире, использующих существующие общественные или частные телефонные компании. Она состоит из трех основных сегментов (космический сегмент, сегмент пользователя, наземный сегмент), взаимодействующих с существующими наземными сетями связи (рис. 4.29).

Космический сегмент системы «Глобалстар» представляет собой группировку из 48 низкоорбитальных спутников со следующими характеристиками:

Число орбит	8
Высота орбит, км	1414
Наклонение, град	52
Число спутников на одной орбите	6
Число космических аппаратов, одновременно обслуживающих территорию России,не менее	4

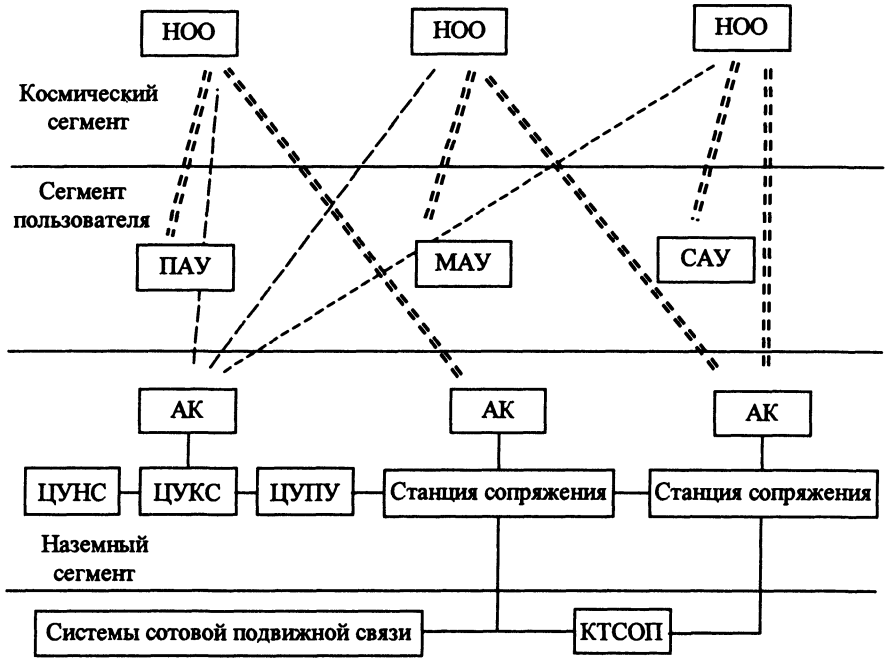


Рис.4.29. Структурная схема системы «Глобалстар»

Каждый спутник содержит антенный комплекс, формирующий 16 лучей, создающих на поверхности Земли зону обслуживания диаметром в несколько тысяч километров, внутри которой возможна коммутация на любую CDMA-несущую с шириной полосы развертывания 1,25 МГц.

Благодаря низкой орбите спутников задержка сигнала и его искажения минимальны. Спутниковая система «Глобалстар» разработана таким образом, что не требует сложных и мощных пользовательских терминалов и наземных станций, что позволяет использовать портативные пользовательские терминалы, сопоставимые по размерам с обычными сотовыми телефонами.

Пользовательский сегмент состоит из портативных (ПАУ), мобильных (МАУ) и стационарных (САУ) терминальных устройств, использующихся для передачи голоса, данных и определения местоположения. Различают одно-, двух- и трехмодовые устройства. Одномодовые устройства используют только для доступа к системе «Глобалстар». Двух- и трехмодовые устройства, кроме доступа к системе «Глобалстар», также используются для доступа к наземным сотовым или другим мобильным радиосетям.

Наземный сегмент состоит из станций сопряжения, центра управления наземным сегментом (ЦУНС), центра управления космическим сегментом (ЦУКС), телекомандного оборудования, сети передачи данных и центров управления поставщиков услуг (ЦУПУ).

Станции сопряжения обеспечивают взаимодействие спутниковой системы подвижной связи и наземных кабельных и мобильных сетей. Большое количество таких станций, установленных по всему миру, гарантирует непрерывное обслуживание пользователей. ЦУНС планирует и контролирует использование ресурсов спутников (каналов, ширины полосы частот и т. п.) станциями сопряжения и взаимодействует с ЦУКС. Наземный центр управляет также сетью передачи данных и станциями сопряжения.

ЦУКС через телекомандные устройства следит за работоспособностью спутников и состоянием их орбит. Он обрабатывает, отображает в реальном времени и проверяет на соответствие параметрам данные телеметрии, поступающие со спутников; в случае несоответствия требованиям выдает отчет об отклонениях. ЦУКС также распределяет и корректирует положение постоянных орбит посредством команд, передаваемых на спутники. Телекомандное оборудование, расположенное на отдельных станциях сопряжения, обеспечивает прием телеметрии и управление спутниковой группировкой. Сеть передачи данных GDN (Globalstar Data Network) – служит для обеспечения связи между частями наземного сегмента системы (управляется и контролируется ЦУНС) и предоставляет широкий спектр возможностей для связи станций сопряжения, ЦУНС, ЦУКС, телекомандного оборудования и делового офиса системы «Глобалстар». Схема организации связи такова, что все вызовы (как местные, так и международные) обрабатываются и коммутируются в наземной станции сопряжения с последующей передачей в телефонную сеть общего пользования (ТФОП). Это обеспечивает связь с абонентскими терминалами «Глобалстар» и другими наземными телефонными и сотовыми сетями, причем перечень услуг, предоставляемых местными операторами связи, не дублируется, а дополняется. Станция сопряжения подключает спутниковую сеть к наземной сети мобильной связи (ССПС), такой, как AMPS или GSM, или непосредственно к АТС местной телефонной сети (ТФОП) посредством стандартного канала E1/T1, поддерживающего различные типы протоколов.

Наземный сегмент на территории России включает три станции сопряжения (Москва, Новосибирск, Хабаровск), обеспечивающей около 98 % охвата территории России южнее 70-й параллели с гарантированным качеством обслуживания. Каждая станция сопряжения связана с сетью общего пользования страны и может быть интегрирована с действующими стационарными и сотовыми сетями России. Каждая станция включает в себя антенную площадку с четырьмя антенными комплексами (АК) (диаметр антенны 5 м), приемопередающую радиоаппаратуру, а также оборудование для маршрутизации и коммутации вызовов (отвечает за земную связь, в том числе за GSM-связь).

Коммутационное и канальное оборудование связано волоконно-оптическими линиями с антенной площадкой. Оно осуществляет передачу и прием сигналов CDMA на промежуточной частоте.

Программно-аппаратный комплекс управления оборудованием станции сопряжения обеспечивает не только визуализацию сети в целом, но и оперативное отслеживание спутников, работающих в данный момент с антеннами станции сопряжения. В число функций программного обеспечения (ПО) управления

входят формирование ежесуточных отчетов о возникающих неисправностях, работе антенных комплексов, активности абонентов и многое другое. Все российские станции сопряжения соединены между собой магистралями (64 кбит/с), принадлежащими «ГлобалТел», по которым и проходит внутренний трафик российского сегмента «Глобалстар».

Система «Глобалстар» для передачи данных в обоих направлениях использует следующий диапазон частот с использованием архитектур CDMA и TDMA/FDMA:

абонентские линии

1610...1621,35 МГц (линия вверх);

2483,5...2500 МГц (линия вниз).

фидерные линии

5091...5250 МГц (линия вверх);

6875...7055 МГц (линия вниз).

Абонентские линии обеспечивают передачу между спутниками и конечными пользователями мобильных терминалов, фидерные линии – передачу сообщений пользователя между спутниками и наземными базовыми станциями.

Сегмент пользователя образуют абонентские терминалы нескольких типов:

- портативные (трубка в руке), аналогичные сотовым телефонам;
- мобильные (устанавливаемые на подвижном средстве);
- стационарные телефонные аппараты, концентраторы, таксофоны.

Выпускаются портативные и мобильные абонентские терминалы, предназначенные как для работы только в системе «Глобалстар», так и для работы в системе «Глобалстар» и сотовых сетях. Это – трехмодовые терминалы, позволяющие работать в системах «Глобалстар», AMPC и CDMA, двухмодовые терминалы для работы в системах «Глобалстар» и GSM и одномодовые терминалы предназначенные только для системы «Глобалстар».

Портативные терминалы (переносимые телефонные трубки) одно-, двух- и трехмодовые имеют излучаемую мощность не более 400 мВт, мобильные (портативный терминал с автомобильным адаптером и внешней антенной) – не более 2 Вт, а стационарные терминалы с дополнительным усилителем и внешней антенной с усилением +7 дБ – не более 3,2 Вт. Телефонные аппараты поддерживают не только основной режим речевой связи, «речевой почты» (хранение речевых сообщений для абонента), услуг по передаче письменных сообщений или срочного вызова в экстренной ситуации, но и передачу телефаксных сообщений Группы 3, передачу данных при пропускной способности до 7,2 кбит/с (работа на несущих частотах в режиме GSM позволяет иметь пропускную способность до 9,6 кбит/с). Это осуществляется через имеющийся последовательный порт ввода/вывода данных, который представляет собой интерфейс с телефаксными аппаратами, компьютерами и другими внешними устройствами.

5. СЕТЕВЫЕ ПРОТОКОЛЫ

Глава посвящена рассмотрению основных сетевых протоколов. В ней рассмотрены вопросы иерархии протоколов и организации протокольных стеков. Основное внимание уделено рассмотрению протоколов стека TCP/IP. Нижний уровень стека TCP/IP представлен протоколами HDLC, LAPD, SLIP и PPP. При описании протокола IP уделено внимание развитию этого протокола при межсетевом взаимодействии в настоящее время. Значительное внимание уделено протоколам маршрутизации RIP и OSPF. Описаны также протоколы TCP и UDP. Приведены краткие сведения о некоторых прикладных протоколах.

5.1. Иерархия протоколов

Современные сети ЭВМ часто строят с использованием нескольких различных базовых технологий. Эта неоднородность возникает либо при объединении уже существовавших ранее сетей, использующих в своих транспортных подсистемах различные протоколы канального уровня, либо при переходе к новым технологиям. Создание сложной структурированной сети, интегрирующей различные базовые технологии, не может осуществляться только средствами одного, например, канального уровня, так как возможностью трансляции протоколов канального уровня обладают далеко не все типы мостов и коммутаторов, к тому же возможности эти ограничены. В частности, в объединяемых сетях должны совпадать максимальные размеры полей данных в кадрах, так как канальные протоколы, как правило, не поддерживают функции фрагментации пакетов.

При объединении нескольких сетей с помощью мостов или коммутаторов также действуют ограничения на топологию: в получившейся сети должны *отсутствовать петли.* Действительно, мост или его функциональный аналог (коммутатор) могут осуществлять доставку пакета адресату только тогда, когда между отправителем и получателем существует единственный путь. В то же время наличие избыточных связей, которые и образуют петли, часто необхо-

димо для лучшей балансировки нагрузки, а также для повышения надежности сети за счет существования альтернативного маршрута в дополнение к основному.

Локальные сети, как правило, являются компонентами больших сетей. В таких сетях одновременно работают несколько различных протоколов, обеспечивающих следующие операции с данными: подготовку, передачу, прием, последующие действия. Работа этих протоколов должна быть скоординирована так, чтобы исключить конфликты или незаконченные операции. Этого можно достичь с помощью разбиения протоколов на уровни по аналогии с моделью OSI.

Стек протоколов (protocol stack) – комбинация согласованных протоколов. Каждый уровень определяет различные протоколы для управления функциями связи или ее подсистемами, и ему присущ свой набор правил. Так же как и уровни в модели OSI, нижние уровни стека протоколов описывают правила взаимодействия оборудования, изготовленного разными производителями, а верхние уровни – правила для проведения сеансов связи и интерпретации приложений. Чем выше уровень, тем сложнее решаемые им задачи и связанные с этими задачами протоколы.

5.2. Стандартные стеки

В компьютерной промышленности в качестве стандартных моделей протоколов разработано несколько стеков. Наиболее важными из них являются:

- набор протоколов ISO/OSI;
- IBM System Network Architecture (SNA);
- Digital DECnet;
- Novell NetWare;
- Apple Apple Talk;
- набор протоколов Интернета, TCP/IP.

Протоколы перечисленных стеков выполняют работу, специфичную для своего уровня. В зависимости от коммуникационных задач, которые возложены на сеть, различают протоколы трех типов: прикладные, транспортные и сетевые.

Прикладные протоколы

Прикладные протоколы работают на верхнем уровне модели OSI. Они обеспечивают взаимодействие приложений и обмен данными между ними. К наиболее популярным прикладным протоколам относятся:

APPC (Advanced Program-to-Program Communication) – одноранговый SNA-протокол фирмы IBM, используемый в основном на AS/400;

FTAM (File Transfer Access and Management) – протокол OSI доступа к файлам;

X.400 – протокол CCITT для международного обмена электронной почтой;

X.500 – протокол CCITT служб файлов и каталогов на нескольких системах;

SMTP (Simple Mail Transfer Protocol) – протокол Интернета для обмена электронной почтой;

FTP (File Transfer Protocol) – протокол Интернета для передачи файлов;

SNMP (Simple Network Management Protocol) – протокол Интернета для мониторинга сети и сетевых компонентов;

Telnet – протокол Интернета для регистрации на удаленных хостах и обработки данных на них;

Microsoft SMBs (Server Message Blocks, блоки сообщении сервера) и клиентские оболочки или редиректоры;

NCP (Novell NetWare Core Protocol) и клиентские оболочки или редиректоры фирмы Novell;

Apple Talk и Apple Share – набор сетевых протоколов фирмы Apple;

AFP (AppteTalk Filling Protocol) – протокол удаленного доступа к файлам фирмы Apple;

DAP (Data Access Protocol) – протокол доступа к файлам сетей DECnet.

Транспортные протоколы

Транспортные протоколы поддерживают сеансы связи между компьютерами и гарантируют надежный обмен данных между ними. К популярным транспортным протоколам относятся:

TCP (Transmission Control Protocol) – TCP/IP-протокол для гарантированной доставки данных, разбитых на последовательность фрагментов:

SPX – часть набора протоколов IPX/SPX (Internetwork Packet Exchange/ Sequential Packet Exchange) для данных, разбитых на последовательность фрагментов, фирмы Novell;

NWLink – реализация протокола IPX/SPX фирмы Microsoft;

NetBEUI [NetBIOS (Network Basic Input/Output System) Extended User Interface – расширенный интерфейс пользователя] – устанавливает сеансы связи между компьютерами (NetBIOS) и предоставляет верхним уровням транспортные услуги (NetBEUI);

ATP (AppleTalk Transaction Protocol), NBP (Name Binding Protocol) – протоколы сеансов связи и транспортировки данных фирмы Apple.

Сетевые протоколы

Сетевые протоколы обеспечивают услуги связи. Они управляют несколькими типами данных: адресацией, маршрутизацией, проверкой ошибок и запросами на повторную передачу. Кроме того, сетевые протоколы определяют правила для осуществления связи в конкретных сетевых средах, например Ethernet или Token Ring. К наиболее популярным сетевым протоколам относятся:

IP (Internet Protocol) – TCP/IP-протокол для передачи пакетов:

IPX (Internetwork Packet Exchange) – протокол фирмы NetWare для передачи и маршрутизации пакетов;

NWLink – реализация протокола IPX/SPX фирмой Microsoft;

NetBEUI – транспортный протокол, обеспечивающий услуги транспортировки данных для сеансов и приложений NetBIOS;

DDP (Datagram Delivery Protocol) – AppleTalk-протокол транспортировки данных.

Протоколы сетевого уровня служат для образования единой транспортной системы, объединяющей несколько сетей с различными принципами передачи информации между конечными узлами. Когда две или более сетей организуют совместную транспортную службу, то такой режим взаимодействия обычно называют *межсетевым взаимодействием* (internetworking). Для обозначения составной сети в англоязычной литературе часто также используется термин *интерсеть* (internetwork или internet).

5.3. Стек протоколов TCP/IP

Стек или набор общих протоколов для разнородной вычислительной среды был разработан по инициативе Министерства обороны США (Department of Defence, DoD) более 20 лет назад для связи экспериментальной сети ARPA с другими сетями. Этот стек получил название TCP/IP по названию двух основных протоколов, входящих в него – Transmission Control Protocol (TCP) и Internet Protocol (IP). В настоящее время TCP/IP – наиболее популярный промышленный стандарт стека протоколов, разработанный для глобальных сетей.

Так как стек TCP/IP был разработан до появления модели взаимодействия открытых систем OSI/ISO, то, хотя он также имеет многоуровневую структуру, соответствие уровней стека TCP/IP уровням модели OSI достаточно условно.

Структура стека протоколов TCP/IP приведена на рис. 5.1. Протоколы TCP/IP делят на 4 уровня. Уровень IV (самый нижний) соответствует физическому и канальному уровням модели OSI. Этот уровень в протоколах TCP/IP не регламентирован, но поддерживает все популярные стандарты физического и канального уровня: для локальных сетей это – Ethernet, Token Ring, FDDI, Fast Ethernet, 100VG-AnyLAN, для глобальных сетей – протоколы соединений «точка-точка» SLIP и PPP, протоколы территориальных сетей с коммутацией пакетов X.25, Frame relay. Обычно при появлении новой технологии локальных или глобальных сетей ее протокол включают в стек TCP/IP за счет разработки соответствующего RFC, определяющего метод инкапсуляции пакетов IP в кадры.

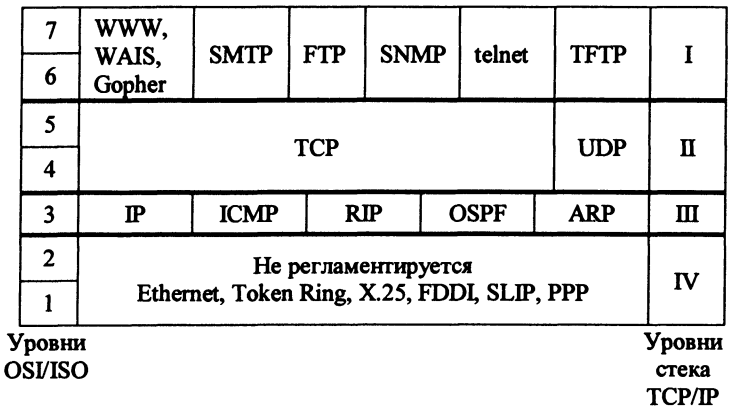


Рис. 5.1. Структура стека протоколов TCP/IP

Уровень III – это уровень межсетевого взаимодействия, осуществляющий передачу пакетов с использованием различных транспортных технологий локальных сетей, территориальных сетей, линий специальной связи и т. п.

В качестве основного протокола уровня межсетевого взаимодействия в стеке TCP/IP используют протокол IP, который хорошо работает в сетях со сложной топологией. Протокол IP – дейтаграммный протокол, т. е. он не гарантирует доставку пакетов до узла назначения.

К уровню межсетевого взаимодействия относятся и все протоколы, связанные с составлением и модификацией таблиц маршрутизации, такие, как протоколы сбора маршрутной информации RIP (Routing Internet Protocol) и OSPF (Open Shortest Path First), а также протокол межсетевых управляющих сообщений ICMP (Internet Control Message Protocol).

Уровень II – основной. На этом уровне функционируют протокол управления передачей TCP (Transmission Control Protocol) и протокол дейтаграмм пользователя UDP (User Datagram Protocol). Протокол TCP обеспечивает надежную передачу сообщений между удаленными прикладными процессами за счет образования виртуальных соединений. Протокол UDP обеспечивает передачу прикладных пакетов дейтаграммным способом, как и IP, но выполняя только функции связующего звена между сетевым протоколом и многочисленными прикладными процессами.

Уровень I (верхний уровень) – прикладной. За долгие годы использования в сетях различных стран и организаций стек TCP/IP накопил большое количество протоколов и сервисов прикладного уровня. К ним относятся такие широко используемые протоколы, как протокол передачи файлов FTP, протокол эмуляции терминала telnet, почтовый протокол SMTP, используемый в электронной почте сети Internet, гипертекстовые сервисы доступа к удаленной информации, такие, как WWW, и многие другие.

5.4. Протоколы IV уровня стека TCP/IP

Протокол HDLC

HDLC – бит-ориентированный протокол управления каналом передачи данных, опубликован стандартом ISO и базовым для построения других протоколов канального уровня (LAP, LAPB, LAPD, LAPX и LLC802.2). Для управления потоком и обнаружения ошибок используется алгоритм «скользящего окна», протокол поддерживает полудуплексную и полнодуплексную передачу, одноточечную и многоточечную конфигурации, а также коммутируемые и некоммутируемые каналы.

Существует *три типа станций* HDLC:

Первичная станция (ведущая) управляет звеном передачи данных (каналом). Несет ответственность за организацию потоков передаваемых данных и восстановление работоспособности звена передачи данных. Эта станция передает кадры команд вторичным станциям, подключенным к каналу. В свою очередь она получает кадры ответа от этих станций. Если канал многоточечный,

главная станция отвечает за поддержку отдельного сеанса связи с каждой станцией, подключенной к каналу.

Вторичная станция (ведомая) работает как зависимая по отношению к первичной станции (ведущей). Она реагирует на команды, получаемые от первичной станции, отсылая кадры ответов. Поддерживает только один сеанс связи с первичной станцией. Вторичная станция не отвечает за управление каналом.

Комбинированная станция сочетает в себе одновременно функции первичной и вторичной станций. Передает как команды, так и ответы и получает команды и ответы от другой комбинированной станции, с которой поддерживает сеанс.

Станции в процессе взаимодействия друг с другом могут находиться в одном из трех логических состояний: логического разъединения, инициализации, передачи информации.

Состояние логического разъединения (LDS). В этом состоянии станция не может вести передачу или принимать информацию. Если вторичная станция находится в нормальном режиме разъединения (NDM – Normal Disconnection Mode), она может принять кадр только после получения явного разрешения на это от первичной станции. Если вторичная станция находится в асинхронном режиме разъединения (ADM – Asynchronous Disconnection Mode), то она может инициировать передачу без получения на это явного разрешения.

Состояние инициализации (IS). Это состояние используется для передачи управления на удаленную вторичную/комбинированную станцию, ее коррекции в случае необходимости, а также для обмена параметрами между удаленными станциями.

Состояние передачи информации (ITS). Вторичной, первичной и комбинированным станциям разрешено вести передачу и принимать информацию пользователя. В этом состоянии станция может находиться в режимах NRM, ARM и ABM.

Режим нормального ответа (NRM – Normal Response Mode) требует, чтобы прежде, чем начать передачу, вторичная станция получила явное разрешение от первичной. После получения разрешения вторичная станция начинает передачу ответа, который может содержать данные. Пока канал использует вторичная станция, может передаваться один или более кадров. После последнего кадра вторичная станция должна снова ждать явного разрешения, прежде чем снова начать передачу. Как правило, этот режим используется вторичными станциями в многоточечных конфигурациях звена передачи данных.

Режим асинхронного ответа (ARM – Asynchronous Response Mode) позволяет вторичной станции инициировать передачу без получения явного разрешения от первичной станции (обычно, когда канал свободен, т. е. в состоянии покоя). Этот режим придает большую гибкость работы вторичной станции. Можно передавать один или несколько кадров данных или управляющую информацию, отражающую изменение состояния (статуса) вторичной станции. С помощью ARM можно уменьшить накладные расходы, поскольку вторичная

станция для передачи данных, не нуждается в опросе. Как правило, такой режим используется для управления соединенными в кольцо станциями или же в многоточечных соединениях с опросом по цепочке. В обоих случаях вторичная станция может получить разрешение от другой вторичной станции и в ответ на него начать передачу. Таким образом разрешение на работу продвигается по кольцу или вдоль соединения.

Асинхронный сбалансированный режим (ABM – Asynchronous Balanse Mode) используют комбинированные станции. Комбинированная станция может инициировать передачу без получения предварительного разрешения от другой комбинированной станции. Этот режим обеспечивает двусторонний обмен потоками данных между станциями, является основным (рабочим) и широко используемым на практике.

Для обеспечения совместимости взаимодействий между станциями, использующих различные процедуры и способных в процессе работы менять свой статус (первичная, вторичная, комбинированная) в протоколе HDLC предусмотрены три способа конфигурирования канала:

- *несбалансированная конфигурация* (UN – Unbalanced Normal) обеспечивает работу одной первичной и одной или большего числа вторичных станций в одноточечной или многоточечной, полудуплексной или полнодуплексной конфигурациях, с коммутируемым и некоммутируемым каналом. Конфигурацию называют несбалансированной потому, что первичная станция отвечает за управление каждой вторичной станцией и за выполнение команд установления режима;
- *симметричная конфигурация* (UA – Unbalanced Asynchronous) была в исходной версии стандарта HDLC и использовалась в первых сетях. Эта конфигурация обеспечивает функционирование двух независимых двухточечных несбалансированных конфигураций станций. Каждая станция обладает статусом первичной и вторичной и, следовательно, логически рассматривается как две станции – первичная и вторичная. Несмотря на то, что станция может работать как в качестве первичной, так и вторичной станции, реальные команды и ответы мультиплексируются в один физический канал. Этот способ в настоящее время используется редко;
- *сбалансированная конфигурация* (BA – Balanced Asynchronous) состоит из двух комбинированных станций, метод передачи – полудуплексный или дуплексный, канал – коммутируемый или некоммутируемый. Комбинированные станции имеют равный статус в канале и могут несанкционированно посылать друг другу трафик. Каждая станция несет одинаковую ответственность за управление каналом.

В протоколе HDLC в качестве протокольного блока данных выступает кадр. Различают кадры трех типов:

- информационного формата (I-кадр). Необходим для передачи данных конечных пользователей между двумя устройствами;

Флаг	Адрес	Управляющее поле	Информационное поле	CRC	Флаг

Рис. 5.2. Структура кадра HDLC

• супервизорного формата (S-кадр). Выполняет управляющие функции, такие, как подтверждение (квитирование) кадров, запрос на повторную передачу кадров и запрос на временную задержку передачи кадров. Фактическое использование супервизорного кадра зависит от режима работы звена (режим нормального ответа, асинхронный сбалансированный режим, асинхронный режим ответа);

• ненумерованного формата (U-кадр). Выполняет управляющие функции. Такой кадр содержит пять двоичных разрядов, что позволяет определить до 32 команд и 32 ответов. Конкретный тип команды и ответа зависит от класса процедуры HDLC.

Структура кадра HDLC. Кадр (рис. 5.2) состоит из пяти или шести полей. Все кадры должны начинаться и заканчиваться полями флага. Поле флага необходимо, чтобы станции, подключенные к звену данных, постоянно контролировали двоичную последовательность флага. Последовательность флага всегда состоит из 01111110.

HDLC является кодопрозрачным протоколом. Он не зависит от конкретного кода (ASCII/IA5 или EBCDIC) при выполнении функции управления каналом. Кроме того, двоичные комбинации управляющих полей обычно занимают в кадре фиксированные разряды. 8-битовая комбинация флага помещается в начале и в конце кадра, чтобы дать возможность приемнику распознать начало и конец кадра. Кроме уникальной флаговой последовательности 01111110 протокол HDLC использует еще два сигнала:

сигнал аварийного завершения (*abort*) состоит из последовательности единиц (от 7 до 14); состояние покоя обозначено последовательностью пятнадцати или большего числа единиц. Сигнал аварийного завершения помещается в конце кадра. Передающая станция посылает этот сигнал, когда возникает исключительная ситуация, требующая восстановления. Вслед за сообщением об аварийном завершении, чтобы поддерживать канал в активном состоянии, и передача могла продолжаться, могут посылаться флаги;

сигнал покоя означает, что канал находится в состоянии покоя. Сигнал покоя используется в полудуплексном сеансе, когда при обнаружении сигнала покоя изменяется направление передачи на противоположное. Фактическое время между передачами кадров по каналу называется *межкадровым временны́м заполнением*. Это временное заполнение сопровождается передачей между кадрами непрерывной последовательности флагов. Флаги могут быть 8-битовыми комбинациями, или может иметь место совмещение последнего «0» предыдущего флага с первым «0» следующего флага.

Для того, чтобы предотвратить вставку в поток данных пользователя комбинации совпадающей с флагом, передающая станция помещает ноль после пяти подряд идущих единиц, встретившихся в любом месте между начальным и конечным флагами кадра. Этот метод называется вставкой битов (bitstaffing).

Приемник постоянно контролирует поток битов. При получении нуля с пятью далее идущими подряд единицами, он анализирует следующий бит. Если это нуль, он удаляет этот бит. Однако если седьмой бит является единицей, приемник аннулирует восьмой бит. Если восьмой бит нуль, то это означает что получена флаговая комбинация 01111110; если же это единица, то получен сигнал покоя или аварийного завершения и выполняются соответствующие действия.

Таким образом, в протоколе HDLC обеспечена кодовая прозрачность и прозрачность по данным. Протоколу безразлично, какие кодовые комбинации находятся в потоке данных. Единственное, что требуется, – это поддерживать уникальность флагов.

Адресное поле определяет первичную или вторичную станции, участвующие в передаче конкретного кадра. Каждой станции присвоен уникальный адрес. В несбалансированной системе адресные поля в командах и ответах содержат адрес вторичной станции. В сбалансированных системах командный кадр содержит адрес получателя, а кадр ответа содержит адрес передающей станции. Адресное поле всегда содержит адрес вторичного узла, задействованного в текущей связи. Поскольку первичный узел является либо источником связи, либо пунктом назначения, то его адрес можно не указывать, так как он заранее известен всем вторичным узлам (рис. 5.3).

Управляющее поле содержит команду и ответы, а также порядковые номера, используемые для отчетности о прохождении данных в канале между первичной и вторичной станциями. Формат и содержание управляющего поля варьируются в зависимости от использования кадра HDLC.

Формат управляющего поля (информационный, супервизорный или ненумерованный) определяет, как это поле кодируется или используется (рис. 5.4).

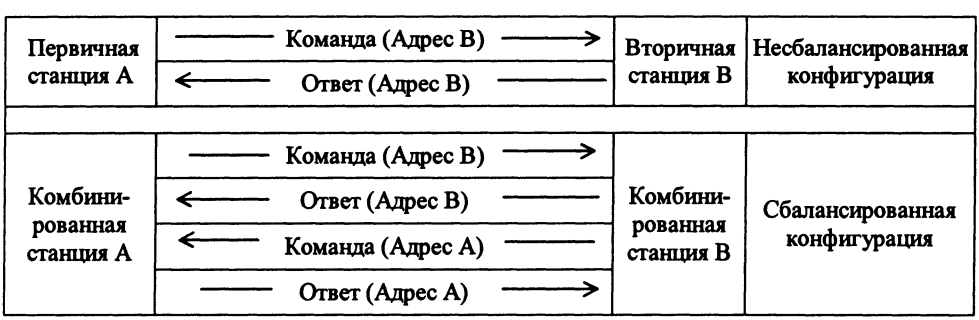


Рис. 5.3. Заполнение поля адреса в различных режимах

1	2	3	4	5	6	7	8	9	10-16	Разряды
0	N(S)							P/F	N(R)	I-формат
1	0	S-коды						P/F	N(R)	S-формат
1	1	U-коды						U-формат		

Рис. 5.4. Формат управляющего поля для различных типов кадра

Информационный кадр (I-кадр) в управляющем поле содержит два порядковых номера: N(S) – порядковый номер передачи (связан с порядковым номером передаваемого кадра); N(R) – порядковый номер приема (означает порядковый номер следующего кадра, который ожидается принимающей станцией). N(R) выступает в качестве подтверждения предыдущих кадров. Эти два поля используются для управления потоком данных и реализуют механизм «скользящего окна».

Пятый двоичный разряд, бит P/F или бит опроса/окончания принимают во внимание, только когда он установлен в «1». Его используют первичная и вторичная станции для выполнения следующих функций:

первичная станция использует бит P для санкционирования передачи кадра статуса от вторичной станции. Бит P также может означать опрос;

вторичная станция отвечает на бит P кадром данных или состояния, а также битом F. Бит F может также означать окончание передачи вторичной станцией в режиме нормального ответа (NRM).

Бит P/F называется битом P, когда его использует первичная станция, и битом F, когда его использует вторичная станция. Только один бит P (ожидающий ответа в виде F бита) может быть активным в канале в любой момент времени. Если некоторый бит P установлен в «1», его можно использовать в качестве контрольной точки, т. е. P = 1 как бы говорит: ответьте мне, потому что я хочу знать ваш статус. Контрольные точки играют большую роль в различных автоматизированных процессах. Они позволяют устранить неопределенность и отменить накопленные транзакции.

Бит P/F может использоваться и интерпретироваться следующим образом:

- в режиме NRM вторичная станция не может вести передачу, пока не будет получена команда с установленным в «1» битом P. Первичная станция может запросить информационные (I) кадры путем посылки кадра с установленным в «1» битом P или посылки некоторых супервизорных (S) кадров (RR, REJ или SREJ) с установленным в «1» битом P;
- в режимах ARM и ABM информационные кадры могут передаваться без запроса с помощью команды, имеющей единичный бит P. Установленный в «1» бит P можно использовать для запроса ответа с установленным в «1» битом F так быстро, насколько это возможно;
- в режимах ARM и ABM производится передача кадра с установленным в «1» битом F вслед за приемом команды с установленным в «1» битом P.

В случае двунаправленной одновременной (полнодуплексной) передачи, когда по получении команды с установленным в «1» битом P передачу ведет вторичная станция, бит F устанавливается в «1» в первом кадре очередного ответа.

Передача кадра с установленным в «1» битом F не требует, чтобы вторичная станция прекратила передачу. Вслед за кадром с установленным в «1» битом F могут быть еще переданы кадры. В режимах ARM и ABM бит F следует просто считать индикатором ответа на предыдущий кадр.

Супервизорный формат (S-кадр) предусматривает четыре команды и ответа: Готов к приему (RR – receive ready), Неприем (REJ – reject), Не готов к приему (RNR – receive not ready), Выборочный неприем (SREJ – selective reject). Назначение этого формата и четырех команд и ответов состоит в выполнении нумерованных (т. е. использующих порядковые номера кадров) супервизорных функций, таких, как подтверждение (квитирование), опрос, временная задержка передачи данных и восстановление после ошибок. Кадры супервизорного формата не содержат информационного поля, но, благодаря наличию порядкового номера приема, их можно использовать для подтверждения приема кадров от передающей станции. Рассмотрим команды и ответы, используемые супервизорным форматом.

Готов к приему (RR) – означает, что первичная или вторичная станция готова принять информационный кадр и/или подтвердить (квитировать) ранее принятые кадры с помощью поля N(R). Если станция до этого, используя команду Не готов к приему (RNR), посылала уведомление о том, что она занята, теперь она использует команду RR для индикации того, что она свободна и готова принять данные. Первичная станция может также использовать команду RR для опроса вторичной станции.

Не готов к приему (RNR) – используется станцией для индикации состояния занятости. Эта команда уведомляет передающую станцию о том, что принимающая станция не способна принять дополнительные поступающие данные. Кадр RNR, используя поле N(R), может подтвердить прием ранее переданных кадров. Состояние занятости может быть сброшено посылкой кадра RR и некоторыми другими кадрами, рассмотренными ниже.

Выборочный неприем (SREJ) используется станцией для запроса повторной передачи единственного кадра, который определен в поле N(R). Как и в случае включающего подтверждения, подтверждение распространяется на все информационные кадры с номерами до N(R) – 1 включительно.

Неприем (REJ) используется для запроса передачи кадров, начиная с кадра, указанного в поле N(R). Подтверждаются все кадры с номерами до N(R) – 1. Кадр REJ используется для реализации метода Возвращение-на-N (Go-Back-N).

Ненумерованный формат в протоколе HDLC предназначен для реализации ненумерованных команд и ответов. Этот формат используется для посылки большинства индикаторов команд и ответов. Структура управляющего поля ненумерованного формата показана (см. рис. 5.4). Ненумерованные команды, в соответствии с выполняемыми функциями, можно разбить на группы:

• команды установки режима: *SNRM, SARM, SABM, SIM, DISC,* (*SNRME, SARME, SABME* для расширенной адресации);
• команды передачи информации: *UI, UP*;
• команды восстановления *RESET*;
• другие команды: *XID, TEST.*

Рассмотрим команды и ответы для ненумерованного формата:

UI (Unnumbered Information – ненумерованная информация) – позволяет осуществлять передачу данных пользователя в ненумерованном кадре (т. е. без порядкового номера).

RIM (Request Initialisation Mode – режим инициализации запроса) – кадр RIM является ответом на команду SIM от вторичной к первичной станции.

SIM (Set Initialisation Mode – установить режим инициализации) – инициализирует сеанс между первичной и вторичной станциями. Ожидаемым ответом является *UA*.

SNRM (Set Normal Response Mode – установить режим нормального ответа) – переводит вторичную станцию в NRM (режим нормального ответа), который предотвращает посылку вторичной станцией несанкционированных кадров. Это означает, что первичная станция управляет всем потоком сообщений в канале.

DM (Disconnect Mode – режим разъединения) – передается вторичной станцией для индикации того, что она находится в режиме логического разъединения.

DISC (Disconnect – разъединить) – команда, передаваемая первичной станцией, переводит вторичную станцию в режим разъединения аналогично нажатию рычага телефонного аппарата.

UA (Unnumbered Ack – ненумерованное подтверждение) – подтверждение *ACK* для установки режима команд (SIM, DISC, RESET). Команда UA также уведомляет об окончании состояния занятости станции.

FRMR (Frame Reject – неприем кадра) – вторичная станция посылает этот кадр, когда она не распознает кадр. Это делается не в случае обнаружения ошибки, указываемой в поле контрольной последовательности, а в более необычных ситуациях. Причина указывается в информационном поле.

Кадр ответа FRMR используют при выполнении следующих условий:
• прием недействительного управляющего поля команды или ответа;
• прием слишком длинного информационного поля;
• прием недействительного поля N(R);
• прием недопустимого информационного поля или супервизорного/ненумерованного кадра неправильной длины.

С помощью кадра FRMR передается значительный объем информации о состоянии (status), при этом информационное поле может содержать следующие данные:
• управляющее поле отвергнутого кадра;
• текущее значение переменных состояния N(S) и N(R) принимающей станции;

- отвергнутый кадр был командой или ответом;
- управляющее поле является недействительным;
- кадр был передан с недопустимым информационным полем;
- информационное поле является слишком длинным;
- порядковые номера являются недействительными.

RD (Request Disconnect – запрос разъединения) – запрос от вторичной станции на логическое разъединение и установление состояния логического разъединения.

XID (Exchage State Identification – идентификация станции при коммутации) – команда запрашивает идентификацию вторичной станции. В системах с коммутацией ее используют для идентификации вызывающей станции.

UP (Unnumbered Polls – ненумированные опросы).

TEST (Test – проверка). Этот кадр используется для санкционирования тестовых ответов от вторичной станции.

SARM (Set Asynchronous Response Mode – установить режим асинхронных ответов) – устанавливает режим, позволяющий вторичной станции вести передачу без опроса со стороны первичной станции. Он переводит вторичную станцию в состояние передачи информации (IS) режима ARM. Поскольку команда SARM устанавливает несбалансированную конфигурацию соединения, SARM должна выдаваться по обоим направлениям передачи:

ООД А посылает: B, DISC; ООД В посылает: B, UA A, DISC;
ООД А посылает: A, UA; ООД В посылает: A, SARM;
ООД А посылает: A, UA B, SARM; ООД В посылает: B, UA

Команды DISC посылаются, чтобы гарантировать полную реинициализацию канала.

SABM (Set Asynchronous Balanced Mode – установить асинхронный сбалансированный режим) – устанавливает режим ABM, в котором станции равноправны. Для передачи не нужен опрос, поскольку каждая станция является станцией комбинированного типа.

SNRME (Set Normal Response Mode Extended – установить расширенный режим нормального ответа) – устанавливает SNRM с двумя дополнительными байтами в управляющем поле.

SABME (Set Asynchronous Balance Mode Extended – установить расширенный асинхронный сбалансированный режим) – устанавливает SABM с двумя дополнительными байтами в управляющем поле.

UP (Unnumbered Poll – ненумерованный опрос) – опрашивает станцию безотносительно к нумерации кадров и квитированию. Если бит опроса установлен в «0», то ответ является необязательным, но такая возможность предоставляется только для одного ответа.

RSET (Reset – сброс переменных) – передающая станция сбрасывает свой порядковый номер передачи N(S), а принимающая станция свой порядковый

номер приема N(R). Эта команда используется для восстановления исходных параметров приемного/передающего окна.

Протокол HDLC, кроме того использует набор системных параметров (T1, N1, N2, K).

Тайм-аут (таймер T1) – с него начинается передача каждого кадра. T1 используют для инициирования повторной передачи, когда таймер переполнен. При выборе периода таймера T1 необходимо учитывать, запускается ли таймер в DCE по началу или по концу кадра. Период таймера T1, по истечении которого можно начинать повторную передачу кадра, в соответствии с процедурами установления и разъединения звена, является системным параметром, подлежащим согласованию с администрацией на некоторый период времени. Для правильной работы процедуры необходимо, чтобы период таймера T1 был больше, чем максимальное время между передачей некоторого кадра (SARM, SABM, DM, DISC, FRMR, I или супервизорной команды) и приемом соответствующего кадра, возвращаемого в качестве отклика на этот кадр (UA, DM или подтверждающий кадр).

Счетчик (N2) – нужен для определения максимального числа повторных передач, которые будут выполнены до того, как переполнится таймер T1. Значение максимального числа передач N2 и повторных передач кадра, вызываемых завершением работы таймера T1 – системный параметр, его необходимо согласовывать с администрацией на некоторый период времени.

Счетчик N1 – максимальное число битов в I-кадре – системный параметр, зависящий от максимальной длины информационных полей, передаваемых через сопряжение DTE/DCE.

Одним из основных параметров протокола HDLC является *максимальное число (K) переданных, но не подтвержденных I-кадров* – число последовательно занумерованных I-кадров, которые в любой момент времени DTE или DCE могут передать без получения подтверждения. Оно должно быть согласовано с администрацией на некоторый период времени и не должно превышать максимального размера окна.

Информационное поле кадра содержит действительные данные пользователя. Информационное поле имеется только в кадре информационного формата. Его нет в кадре супервизорного или ненумерованного формата.

Поле контрольной последовательности кадра (CRC) необходимо для обнаружения ошибок передачи между двумя станциями звена данных. Вычисление CRC осуществляется методом циклического кодирования с производящим полиномом $X^{16} + X^{12} + X^5 + 1$ в соответствии с рекомендацией МККТТ V.41. Это позволяет обнаруживать всевозможные картежи ошибок длиной не более 16 разрядов, вызываемые одиночной ошибкой, а также 99,9984 % всевозможных более длинных кортежей ошибок.

Протокол LAPD

Протокол LAPD (Link Access Procedure on the D-channel) управляет потоком кадров, передаваемых по D-каналу, и предоставляет информацию, необходимую для управления потоком и исправления ошибок. Спецификации протокола как базового, так и первичного доступа определены в рекомендациях ITU-T 1.440 (основные аспекты) и 1.441 (подробные спецификации). Эти же рекомендации в серии Q имеют номера Q.920 и Q.921. Обмен информацией на уровне LAPD осуществляется посредством информационных блоков, называемых кадрами. Форматы и процедуры LAPD основаны на протоколе управления звеном передачи данных высокого уровня HDLC (High-level Data-Link Control procedures), первоначально определенном Международной организацией по стандартизации ISO.

Структура кадра LAPD. Кадры содержат либо команды на выполнение действий, либо ответы, сообщающие о результатах выполнения команд, что определяется специальным битом идентификации команда/ответ C/R. Общий формат кадров LAPD показан на рис. 5.5.

Каждый кадр начинается и заканчивается однобайтовым *флагом*. Комбинация флага (01111110) такая же, как в HDLC. Подмена флага любым другим полем кадра исключена благодаря процедуре «битстаффинга» (bit-stuffing).

Адресное поле (байты 2 и 3) кадра содержит идентификатор точки доступа к услуге SAPI (Service Access Point Identifier) и идентификатор терминала TEI (Terminal EquIPment Identifier). Это поле используется для маршрутизации кадра к месту его назначения. Эти идентификаторы определяют соединение и терминал, к которым относится кадр.

Идентификатор пункта доступа к услуге SAPI занимает 6 бит в адресном поле и фактически указывает, какой логический объект сетевого уровня должен анализировать содержимое информационного поля. Например, SAPI может указывать, что содержимое информационного поля относится к процедурам управления соединениями в режиме коммутации каналов или к процедурам пакетной коммутации. Рекомендацией Q.921 определены значения SAPI (табл. 5.1).

Рис. 5.5. Формат кадра LAPD

Таблица 5.1. **Значения SAPI**

Значения SAPI	Функция
0	Управление соединением ISDN (коммутация каналов)
1	Пакетная коммутация по Q.931
16	Пакетная коммутация X.25
63	Управление уровнем 2

Идентификатор TEI указывает терминальное оборудование, к которому относится сообщение. Код TEI=127 (1111111) определяет на вещательную (циркулярную) передачу информации всем терминалам, связанным с данной точкой доступа. Остальные значения (0 – 126) используют для идентификации терминалов. Диапазон значений TEI (табл. 5.2) разделяется между теми терминалами, для которых TEI назначает сеть (автоматическое назначение TEI), и теми, для которых TEI назначает пользователь (неавтоматическое назначение TEI).

Таблица 5.2. **Значения TEI**

Значения TEI	Назначение
0 – 63	Неавтоматическое назначение TEI
64 –126	Автоматическое назначение TEI
127	Вещательный режим

Бит идентификации команды/ответа C/R (Command/Response bit) в адресном поле перенесен в протокол LAPD из протокола X.25. Этот бит устанавливается LAPD на одном и обрабатывается на противоположном конце звена. Значение C/R (табл. 5.3) классифицирует каждый кадр как командный или как кадр ответа. Если кадр сформирован как команда, адресное поле идентифицирует получателя, а если кадр является ответом, адресное поле идентифицирует отправителя. Отправителем или получателем могут быть как сеть, так и терминальное оборудование пользователя.

Таблица 5.3. **Значение C/R**

Кадр	Кадры, передаваемые	Кадры, передаваемые терминалом
Командный	C/R=1	C/R=0
Ответа	C/R=0	C/R=1

Бит расширения адресного поля EA (Extended address bit) служит для гибкого увеличения длины адресного поля. Бит расширения в первом байте адреса, имеющий значение 0, указывает на то, что за ним следует другой байт. Бит расширения во втором байте, имеющий значение 1, указывает, что этот второй

байт в адресном поле является последним. Именно такой вариант приведен на рис. 5.1. Если впоследствии возникнет необходимость увеличить размер адресного поля, значение бита расширения во втором байте может быть изменено на 0, что будет указывать на существование третьего байта. Третий байт в этом случае будет содержать бит расширения со значением 1, указывающим, что этот байт является последним. Увеличение размера адресного поля, таким образом, не влияет на остальную часть кадра.

Два последних байта в структуре кадра содержат 16-битовое поле *проверочной комбинации* кадра FCS (Frame check sequence) и генерируются уровнем звена данных в оборудовании, передающем кадр. Это поле позволяет протоколу LAPD обнаруживать ошибки в полученном кадре. В поле FCS передается 16-битовая последовательность, биты которой формируются как дополнение для суммы (по модулю 2), в которой:

а) первым слагаемым является остаток от деления (по модулю 2) произведения $x^k(x^{15} + x^{14} + ... + x + 1)$ на образующий полином $(x^{16} + x^{12} + x^5 + 1)$, где k – число битов кадра между последним битом открывающего флага и первым битом проверочной комбинации, исключая биты, введенные для обеспечения прозрачности;

б) вторым слагаемым является остаток от деления (по модулю 2) на этот образующий полином произведения x^{16} на полином, коэффициентами которого являются биты кадра, расположенные между последним битом открывающего флага и первым битом проверочной комбинации, исключая биты, введенные для обеспечения прозрачности.

Обратное преобразование выполняется уровнем звена данных в оборудовании, принимающем кадр, с тем же образующим полиномом для адресного поля, полей управления, информационного и FCS. Протокол LAPD использует соглашение, по которому остаток от деления (по модулю 2) произведения x^{16} на полином, коэффициентами которого являются биты перечисленных полей и FCS, всегда составляет 0001110100001111 (десятичное 7439), если на пути от передатчика к приемнику никакие биты не были искажены. Если результаты обратного преобразования соответствуют проверочным битам, кадр считается переданным без ошибок. Если же обнаружено несоответствие результатов, это означает, что при передаче кадра произошла ошибка.

Поле управления указывает тип передаваемого кадра и занимает в различных кадрах один или два байта. Существует три типа форматов, определяемых полем управления: передача информации с подтверждением (I-формат), передача команд, реализующих управляющие функции (S-формат), и передача информации без подтверждения (U-формат). В табл. 5.4 приведены сведения об основных типах кадров протокола LAPD.

Таблица 5.4. **Основные типы кадров LAPD**

Формат	Команда	Ответ	Описание
I-кадры	Информация	–	Используется в режиме с подтверждением для передачи нумерованных кадров, содержащих информационные поля с сообщениями верхних уровней.
S-кадры	К приему готов (RR)	К приему готов (RR)	Используется для указания готовности встречной стороны к приему I-кадра или для подтверждения ранее полученных I-кадров
	К приему не готов (RNR)	К приему не готов (RNR)	Используется для указания неготовности встречной стороны к приему I-кадра
	Отказ/переспрос (REJ)	Отказ/переспрос (REJ)	Используется для запроса повторной передачи I-кадра
U-кадры	Ненумерованная информация (UI)	–	Используется в режиме передачи без подтверждения
	–	Отключено (DM)	
	Установка расширенного асинхронного балансного режима (SABME)	–	Используется для начальной установки режима с подтверждением
	–	Отказ кадра (FRMR)	
	Разъединение (DISC)	–	Используется для прекращения режима с подтверждением
	–	Ненумерованное подтверждение (UA)	Используется для подтверждения приема команд установки режима, например, SABME, DISC

Рассмотрим эти типы несколько подробнее.

Информационный кадр (I-кадр) – с его помощью организуют передачу информации сетевого уровня между терминалом пользователя и сетью. Этот кадр содержит информационное поле, в котором помещено сообщение сетевого уровня. Поле управления I-кадра содержит порядковый номер передачи (N/S), который увеличивается на 1 (по модулю 128) для каждого передаваемого кадра. При подтверждении приема I-кадров в поле управления вводится порядковый номер приема (N/R).

Управляющий кадр (S-кадр) необходим для поддержки функций управления потоком и запроса повторной передачи. S-кадры не имеют информационного поля. Например, если сеть временно не в состоянии принимать I-кадры, пользователю посылается S-кадр «к приему не готов» (RNR). Когда сеть снова может принимать I-кадры, она передает другой S-кадр – «к приему готов» (RR). S-кадр также можно использовать для подтверждения в этом случае он содержит порядковый номер приема, а не передачи.

Управляющие кадры передают как командные или как кадры ответа.

Ненумерованный кадр (U-кадр). Среди ненумерованных кадров имеется кадр ненумерованной информации (UI), единственный, содержащий информационное поле и несущий сообщение сетевого уровня. U-кадры используют для передачи информации в режиме без подтверждения и некоторых административных директив. Чтобы транслировать сообщение ко всем терминалам, подключенным к шине S-интерфейса, станция передает кадр UI с TEI = 127. Поле управления U-кадров не содержит порядковых номеров.

Информационное поле предусмотрено в кадрах только некоторых типов. В нем заключена информация сетевого уровня, сформированная одной системой, например, терминалом пользователя, которую необходимо передать другой системе, например сети. Информационное поле может быть пропущено, если кадр не имеет отношения к конкретной коммутируемой связи (например, в управляющих кадрах, S-формат). Если кадр относится к канальному уровню и сетевой уровень не участвует в его формировании, соответствующая информация включается в поле управления.

Биты P/F (poll/final) поля управления идентифицируют группу кадров (см. табл. 5.4), что также заимствовано из спецификаций протокола HDLC. Путем установки в «1» бита P в командном кадре функции LAPD на одном конце звена данных указывают функциям LAPD на противоположном конце звена на необходимость ответа управляющим или ненумерованным кадром. Кадр ответа с F = 1 указывает, что он передается в ответ на принятый командный кадр со значением P = 1. Оставшиеся биты байта 4 идентифицируют конкретный тип кадра в пределах группы.

Передача с подтверждением. Этот способ используют для передачи информационных кадров только в соединениях звена данных, имеющих конфигурацию «точка-точка». Он обеспечивает исправление ошибок путем повторной передачи и доставку не содержащих ошибок сообщений в порядке очередности.

Поле управления информационного кадра имеет подполя «номер передачи» N(S) и «номер приема» N(R). Эти подполя аналогичны одноименным полям в HDLC. Протокол LAPD присваивает по модулю 128 возрастающие порядковые номера передачи N(S) последовательно передаваемым информационным кадрам. Он также записывает передаваемые кадры в буфер повторной передачи и хранит их в буфере до получения положительного подтверждения их приема.

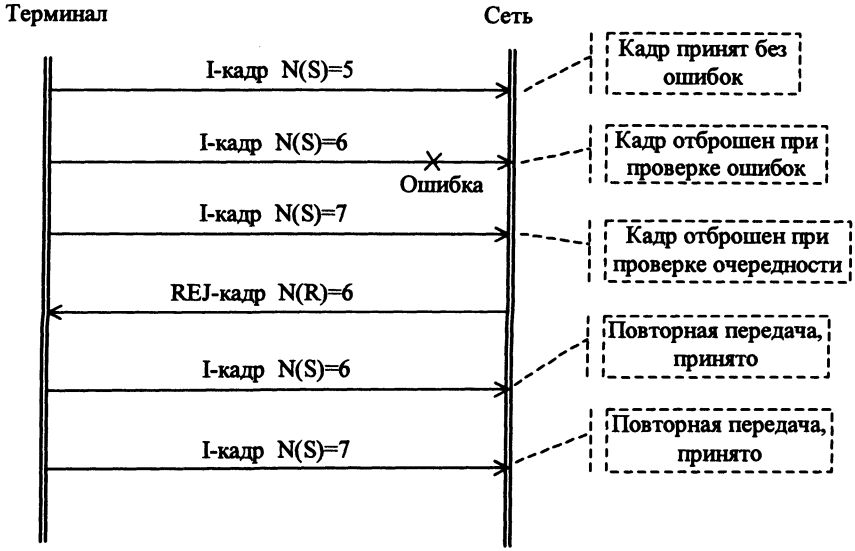


Рис. 5.6. Исправление ошибок в информационном кадре

Рассмотрим передачу информационных кадров с исправлением ошибок от терминала к сети (рис. 5.6). Все поступающие в сеть кадры проверяются на наличие ошибок, а затем в свободных от ошибок информационных кадрах проверяется порядковый номер. Если значение N(S) выше (по модулю 128) на единицу, чем N(S) последнего принятого информационного кадра, новый кадр считается следующим по порядку и поэтому принимается, а его информационное поле пересылается конкретной функции сетевого уровня. После этого сеть подтверждает прием информационного кадра своим исходящим кадром с номером приема N(R), значение которого на единицу больше (по модулю 128), чем значение N(S) в последнем принятом информационном кадре.

Предположим, что последний принятый информационный кадр имел номер N(S) = 5 и что информационный кадр с номером N(S) = 6 передан с ошибкой, в результате которой отбракован функциями LAPD на стороне сети. Следующий информационный кадр с N(S) = 7 успешно проходит проверку на ошибки, но поступает в сеть с нарушением очередности и отбрасывается ею при проверке порядка следования. Тогда сеть передает кадр отказа (REJ) с номером N(R) = 6, который запрашивает повторную передачу информационных кадров из буфера повторной передачи терминала, начиная с кадра с N(S) = 6. Сетевая сторона продолжает отбрасывать информационные кадры при проверке их на порядок следования, пока не примет повторно переданный кадр с номером N(S) = 6.

Нумерация кадров при передаче с подтверждением – одна из важнейших функций протокола LAPD. При выполнении этой процедуры важное значение имеет параметр k – число неподтвержденных квитируемых кадров. Передатчик должен прекратить работу, когда разница между его собственным

значением N(S) (числом переданных кадров I) и значением N(R) (числом подтвержденных кадров I) превысит параметр, обозначаемый *k*. Значение *k* устанавливается в соответствии со спецификой использования звена и скоростью передачи в нем: $k = 1$ – для сигнализации базового доступа BRA при скорости *D*-канала 16 кбит/с, $k = 3$ – для пакетной передачи при скорости 16 кбит/с, $k = 7$ – для сигнализации первичного доступа PRA при скорости *D*-канала 64 кбит/с.

Два потока сообщений от терминала к сети и в обратном направлении для соединения «точка-точка» независимы друг от друга и от потоков сообщений в других соединениях «точка-точка» в том же *D*-канале. В *D*-канале с *n* соединениями типа «точка-точка» могут присутствовать $2n$ независимых последовательностей N(S)/N(R).

Процедура подтверждаемой передачи информации (рис. 5.7). Рассмотрим случай, когда необходимо начать передачу информации уровня 3 от терминала пользователя к сети. Инициатором данной процедуры является уровень 3 на стороне пользователя, который выдает примитив запроса соединения DL_ESTABLISH. По этому запросу уровень 2 на стороне пользователя формирует управляющий кадр установки расширенного асинхронного балансного режима (SABME – Set Asynchronous Balanced Mode Extended). Кадр SABME пересылается к сети через уровень 1. При получении кадра SABME уровнем 2 на стороне сети проверяются условия, необходимые для установки режима подтверждаемой передачи информации (например, чтобы убедиться, что соот-

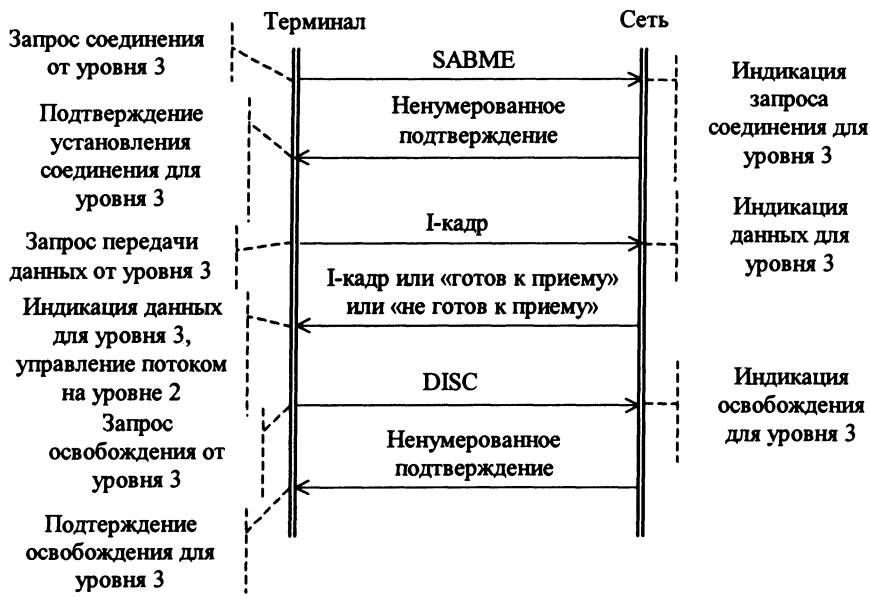


Рис. 5.7. Процедура подтверждаемой передачи

ветствующее оборудование доступно). Если все условия выполнены, уровень 2 на стороне сети посылает уровню 3 примитив индикации запроса соединения, чтобы указать, что устанавливается режим подтверждаемой передачи информации. Средствами уровня 2 сеть возвращает пользователю ненумерованное подтверждение. При получении этого подтверждения терминалом пользователя на уровень 3 передается примитив подтверждения установления соединения, указывающий, что можно начинать подтверждаемую передачу информации. Теперь между пользователем и сетью можно осуществить передачу информации с помощью I-кадров.

Эта информация направляется уровнем 3 к уровню 2 в примитиве запроса передачи данных DL_DATA. Данные помещаются в информационное поле I-кадра и передаются от пользователя к сети через уровень 1. При получении уровнем 2 на стороне сети I-кадра данные извлекаются из информационного поля и передаются к уровню 3 в примитиве индикации приема данных. В зависимости от содержимого полученного I-кадра сеть посылает в ответ пользователю либо I-кадр, либо управляющий кадр готовности к приему. Оба кадра содержат подтверждение, что I-кадр от пользователя был успешно принят.

Каждый I-кадр содержит в поле управления порядковые номера передачи и приема. Процедура обнаружения потерь работает в обоих направлениях. В качестве примера на рис. 5.6 была рассмотрена передача необходимого сетевому уровню числа информационных кадров, включая передачу кадров 5, 6 и 7. Когда обмен I-кадрами, показанный на рис. 5.6, заканчивается, происходит посылка команды разъединения DISC, за которой следует ответ DM, подтверждающий разъединение. На рис. 5.7 уровень 3 на стороне пользователя отправляет уровню 2 примитив запроса освобождения DL_RELEASE, а уровень 2 формирует кадр разъединения, который передается через уровень 1 уровню 2 на стороне сети. При получении кадра разъединения уровнем 2 на стороне сети уровню 3 выдается примитив индикации освобождения, а пользователю возвращается кадр ненумерованного подтверждения. При получении кадра ненумерованного подтверждения уровнем 2 на стороне пользователя уровню 3 выдается примитив подтверждения освобождения для завершения процедуры освобождения.

Передача неподтверждаемых сообщений. Управляющие кадры S и ненумерованные кадры U не содержат подполя N(S). Они принимаются получателем, если получены без ошибок, и на них не отправляется подтверждение. Управляющие кадры содержат поле N(R) для подтверждения принятых информационных кадров.

Ненумерованные информационные кадры UI не содержат ни поля N(S), ни поля N(R), поскольку они передаются в вещательном режиме с TEI = 127, а возможность координировать порядковые номера передачи и приема для групповых функций во всех терминалах, подключенных к одному S-интерфейсу, отсутствует.

Процедура неподтверждаемой передачи информации. Рассмотрим случай, когда необходима передача информации от функций уровня 3 на сторо-

не сети к функциям уровня 3 в терминале пользователя. Функции уровня 3 на стороне сети передают к уровню 2 примитив запроса передачи данных без подтверждения DL_UNIT DATA. Уровень 2 формирует кадр ненумерованной информации (UI – Unnumbered Information), содержащий в информационном поле информацию, которую надо передать. Этот кадр и передается через уровень 1 к функциям уровня 2 в терминале пользователя. Если необходима вещательная (циркулярная) передача кадра всем терминалам, TEI в адресном поле присваивается значение 127. Если же обращение происходит к одному определенному терминалу, т.е. необходим режим «точка-точка», тогда TEI присваивается значение от 0 до 126, совпадающее с TEI, назначенным для этого терминала, например, TEI = 7. При получении кадра UI терминалом пользователя информация, содержащаяся в информационном поле, доставляется из уровня 2 в уровень 3 с помощью примитива индикации приема данных без подтверждения.

При такой неподтверждаемой передаче информации в уровне 2 отсутствует процедура защиты от ошибок. Следовательно, решение о логическом восстановлении кадра в случае его потери или искажения возложено на функции уровня 3.

Рассмотрим подробнее использование управляющих кадров: кадр готовности к приему RR, сообщающий о готовности принимать информационные кадры; кадр неготовности к приему RNR, сообщающий о том, что принимать информационные кадры временно нельзя, но прием управляющих кадров возможен; кадр отказа REJ, указывающий, что поступивший информационный кадр отброшен. На рис. 5.8 показаны несколько примеров, которые иллюстрируют использование битов C/R, P и F.

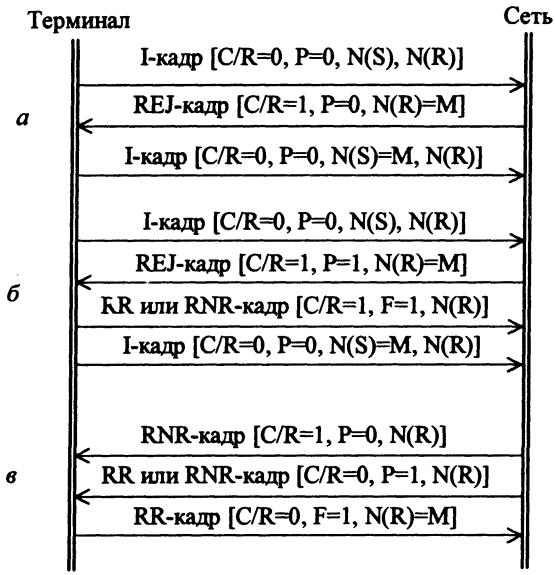


Рис. 5.8. Примеры процедур контроля звена передачи данных

На рис. 5.8, *а* уровень 2 на стороне сети получил информационный кадр с нарушением порядка очередности и отбрасывает его с помощью команды REJ, в которой бит P имеет значение 0 (подтверждения не требуется). N(R) = M указывает, что последний принятый информационный кадр имел N(S) = M – 1. Терминал повторяет передачу информационных кадров из своего буфера повторной передачи, начиная с кадра, для которого N(S) = M.

На рис. 5.8, *б* рассмотрена та же ситуация, за исключением того, что в командном кадре REJ бит P =1. Этим передается указание терминалу пользователя подтвердить кадр. Терминал пользователя сначала передает кадр ответа RR или RNR (C/R = 1, F = 1), а затем начинает повторную передачу информационных кадров.

На рис. 5.8, *в* сетевая сторона указывает с помощью командного кадра RNR, что она не может принимать информационные кадры. Сторона пользователя приостанавливает передачу информационных кадров и запускает таймер. Если терминал получает кадр RR до срабатывания таймера, то он возобновляет передачу или повторную передачу информационных кадров. Если таймер сработал, а кадр RR не получен, терминал пользователя передает кадр команды (C/R = 1) с P = 1. Этим дается указание сетевой стороне передать, в свою очередь, командный кадр. В данном примере сетевая сторона отвечает кадром RR, указывая, что она готова снова принимать информационные кадры и что номер последнего принятого кадра N(S) = M – 1. Затем сторона терминала возобновляет передачу информационных кадров, начиная ее кадром с номером N(S) = M. Если ответом сетевой стороны будет кадр RNR, то сторона пользователя перезапустит свой таймер и снова будет ожидать кадр RR. Если сетевая сторона остается неготовой к приему после нескольких срабатываний таймера, то сторона пользователя передает решение вопроса в более высокую инстанцию – к соответствующей функции сетевого уровня.

Процедуры управления TEI. Для протокола LAPD определены процедуры управления TEI, т. е. процедуры его назначения, контроля и отмены. Для соединений «точка-точка» в терминале запоминается «свой» TEI и проверяется TEI в поле адреса принимаемых кадров, чтобы определить, не предназначен ли кадр этому терминалу. Терминал также вводит свой TEI в адресные поля передаваемых им кадров.

Терминалы (TE) подразделяются на терминалы с неавтоматическим и автоматическим механизмом назначения TEI. TE первого типа ориентированы на длительное подключение к одной цифровой абонентской линии, с постоянно активным физическим уровнем. Эти терминалы имеют ряд переключателей, положение которых определяет значение TEI. Переключатели устанавливает технический персонал при инсталляции TE, и их положение не меняется, пока TE подключен к этой цифровой абонентской линии. TE такого типа имеют значения от 0 до 63.

Автоматическое присвоение TEI применяется в тех случаях, когда используются процедуры активизации/деактивизации физического уровня интерфейса «пользователь-сеть» (при деактивизации физического уровня TEI сбрасывается) или когда терминальное оборудование работает непостоянно. Менять значение TEI вручную при каждом перемещении неудобно, поэтому для мобильных TE применяется автоматическое назначение TEI (в диапазоне 64–126), а также его проверка и отмена, для чего и используются упомянутые выше процедуры управления TEI. Этими процедурами предусмотрены сообщения следующих типов:

Рис. 5.9. Сообщение управления TEI

Запрос ID. Сообщение передается мобильным TE, когда необходимо, чтобы сеть назначила для него TEI.

ID назначен. Это ответ сети на запрос ID. Он содержит назначенный TEI.

Отказ в назначении ID. Это ответ сети, отвергающий запрос ID.

Запрос проверки ID. Это команда от сети для проверки назначенного значения TEI.

Ответ проверки ID. Это ответ мобильного TE на запрос-проверки ID.

Отмена ID. Эта команда передается от сети к TE, чтобы отменить назначенный ранее TEI.

Все сообщения передаются в кадрах UI с SAPI = 63. Информационное поле кадров UI показано на рис. 5.9. Код в байте 1 указывает, что это сообщение управления TEI. Код типа сообщения находится в байте 4 (табл. 5.5). Сообщение содержит параметры Ri (ссылочный номер) и Ai (индикатор действия).

Таблица 5.5. **Коды типа сообщения**

Тип сообщения	Направление TE сеть	Код типа сообщения	Ссылочный номер Ri	Индикатор действия Ai
Запрос ID	→	0000 0001	0 – 65535	127
ID назначен	←	0000 0010	– „ –	64 – 126
Отказ в значении ID	←	00000011	– „ –	64 – 127
Запрос проверки ID	←	0000 0100	–	0 – 127
Ответ проверки ID	→	0000 0101	0 – 65535	0 – 126
Отмена ID	←	00000110	–	0 – 127
Верификация ID	→	00000111	–	0 – 126

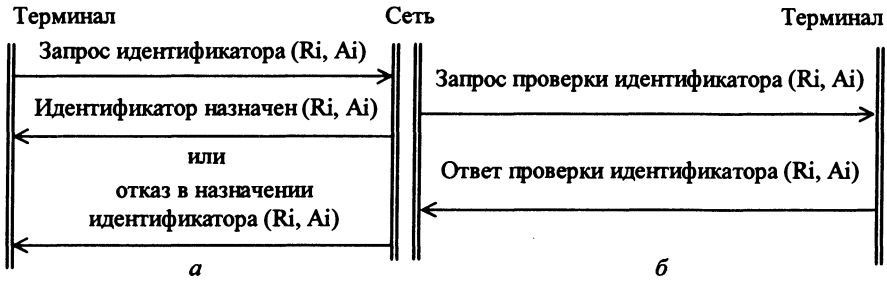


Рис. 5.10. Процедура управления TEI:
а – назначение; *б* – проверка

Процедура назначения TEI дает возможность оборудованию пользователя, имеющему категорию «мобильный», получить от сети номер TEI, который можно использовать при последующих соединениях. Процедура назначения показана на рис. 5.10, а. Когда мобильный TE подсоединяется к S-интерфейсу, он автоматически посылает запрос ID. Поскольку терминальное оборудование не имеет TEI, то, чтобы идентифицировать себя, оно генерирует произвольный ссылочный номер (Ri). TE может запросить сеть назначить для него конкретный TEI, указав этот TEI в поле Ai, или оставить право выбора TEI за сетью, поместив в это поле Ai = 127.

Для каждой цифровой абонентской линии сеть поддерживает список мобильных TEI в диапазоне 64 – 126. При получении от некоторого S-интерфейса сообщения «запрос ID» сеть обращается к соответствующему списку. Если она может назначить TEI, то по данной шине S-интерфейса в вещательном режиме передается сообщение «ID назначен», в котором значение Ri копируется из сообщения «запрос ID», а назначенный TEI помещается в поле Ai. Все TE, подключенные к этой S-шине, проверяют сообщение, но только TE, который послал запрос, опознает свое Ri и воспринимает назначенный TEI. Такая процедура позволяет двум или более TE, подключенным к одной и той же S-шине, посылать запросы ID одновременно.

Если сеть не может удовлетворить запрос ID из-за того, что запрошенный TEI уже есть в списке назначенных для данного интерфейса или из-за того, что все TEI в диапазоне 64 – 126 уже назначены, она передает по S-шине этого интерфейса в вещательном режиме сообщение «отказ в назначении ID», снова копируя Ri из принятого запроса. После этого TE информирует своего пользователя о том, что его запрос на назначение TEI был отвергнут.

Процедура проверки TEI позволяет сети проконтролировать список мобильных TEI, назначенных для конкретного интерфейса (рис. 5.10, *б*). Сеть передает этому интерфейсу в вещательном режиме сообщение «запрос проверки ID», поместив в поле Ai проверяемый TEI, а в поле Ri – нулевое значение. При этом сеть запускает таймер на 200 мс. Если среди подключенных к данному интерфейсу найдется TE, имеющий TEI, который совпадает с Ai, он отвечает сообщением «ответ проверки ID», содержащим произвольно выбранное Ri и принятое Ai.

В нормальных условиях сеть принимает до срабатывания таймера одно сообщение «ответ проверки ID», что указывает на наличие единственного ТЕ с данным TEI. Если таймер сработал, а ответ не получен, сеть повторяет запрос проверки ID и перезапускает таймер. Если таймер снова срабатывает до получения ответа, сеть считает, что данный TEI больше не используется, удаляет его из списка TEI, назначенных для данного интерфейса, и составляет отчет для обслуживающего персонала.

В случае получения сетью более одного ответа на «Запрос проверки ID», это означает, что один и тот же TEI ошибочно присвоен более чем одному ТЕ. В этом случае сеть передает в вещательном режиме команду «отмена ID» с указанием в поле Ai отменяемого TEI. Те терминалы, TEI которых согласуются с Ai, прекращают передачу и прием кадров и уведомляют своего пользователя об отмене TEI. Если сеть решает, что значение TEI должно быть отменено, вызывается *процедура отмены*. Сеть формирует кадр, содержащий тип сообщения и поле индикатора действия, где помещается значение TEI, которое должно быть отменено. Для уменьшения риска потери такой кадр посылается дважды.

Протокол SLIP

Протокол SLIP (Serial Line IP) стал первым промышленным стандартом де-факто, который позволил устройствам, соединенным последовательным низкоскоростным интерфейсом связи, работать по протоколам TCP/IP. Этот Internet-протокол разрешает в качестве линий связи использовать обычные телефонные линии.

Протокол был создан в начале 80-х годов и согласно RFC-1055 впервые был включен в качестве средства доступа к IP-сети в пакет фирмы 3COM – UNET. В 1984 г. протокол SLIP был встроен Риком Адамсом (Rick Adams) в операционную систему 4.2 Berkley Unix. Позднее SLIP был поддержан в других версиях Unix и реализован в программном обеспечении для ПК.

Ввиду своей функциональной простоты, SLIP использовался и используется в основном на коммутируемых линиях связи, которые не характерны для ответственных и скоростных сетевых соединений. Тем не менее, коммутируемый канал отличается от некоммутируемого только более низким качеством и необходимостью выполнять процедуру вызова абонента, поэтому SLIP вполне применим и на выделенных каналах.

Протокол SLIP выполняет единственную функцию – он позволяет в потоке бит, которые поступают по выделенному (или коммутируемому) каналу, распознать начало и конец IP-пакета. Другие протоколы сетевого уровня SLIP не поддерживает.

Программное обеспечение, реализующее работу с протоколом SLIP, принимает символы, приходящие с устройства последовательной передачи данных (модема, последовательного порта и т. д.); рассматривает и толкует их как составляющие IP-пакета; укладывает полученные данные в полнокровный нормальный IP-пакет и передает этот пакет далее – соответствующей програм-

ме, которая обрабатывает IP-пакеты, например модулю TCP. На обратном пути SLIP получает от программы (сетевого уровня), посылающей IP-пакеты, IP-пакет, вычленяет его содержимое, соответствующим образом переформатирует, затем делит на символы и отправляет его через устройство последовательной передачи по последовательной линии в сеть – соседнему узлу Internet.

Структура кадра протокола SLIP. Протокол SLIP предназначен для передачи IP-пакетов через асинхронные линии связи. Поскольку асинхронная передача является байт-ориентированной, то перед транспортировкой средствами SLIP пакет разделяется на октеты (байты), которые передаются один за другим.

Как известно, в сети Ethernet IP-пакет может иметь длину до 1500 байт, что обусловливает необходимость его сегментации – разбиения на более короткие пакеты. SLIP делает это довольно примитивно. Он не анализирует поток данных и не выделяет какую-либо информацию в этом потоке. Для распознавания границы IP-пакетов, протокол SLIP предусматривает использование специального символа END, значение которого в шестнадцатеричном представлении равно $(C0)_h$. Для разделения SLIP-кадров между ними вставляется служебный байт-разделитель – символ ESC $(DB)_h$.

Применение специального символа может породить конфликт: если байт пересылаемых данных тождественен символу END, то он будет ошибочно определен как признак конца пакета. Чтобы такой байт, встретившийся внутри IP-пакета, не воспринимался как разделитель, предусмотрен механизм вставки байта (byte staffing).

Таким образом, собственно служебной информации в протоколе SLIP довольно мало: на IP-пакет добавляется один байт-разделитель (между пакетами они не дублируются), а иногда появляется несколько дополнительных байтов, вставляемых по процедуре вставки байта.

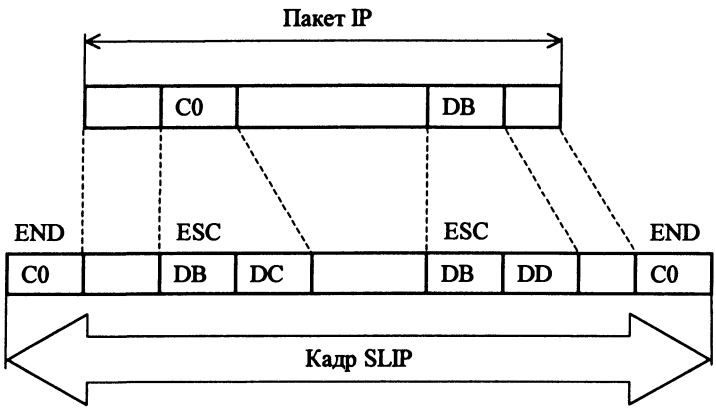


Рис. 5.11. Соответствие между блоками данных протоколов IP и SLIP

Стандарт не определяет фиксированный размер SLIP-кадра, поэтому любой SLIP-интерфейс имеет специальное поле, в котором пользователь должен указать эту длину. Однако в конкретных реализациях максимальный размер SLIP-кадра часто оказывается ограниченным до очень небольшого значения (от 256 до 1006 байт). Данное ограничение связано с первой реализацией протокола SLIP в соответствующем драйвере для Berkley Unix, и его соблюдение необходимо для поддержки совместимости разных реализаций SLIP (большинство современных реализаций поддерживают эту длину и позволяют администратору самому установить его размер, а по умолчанию принимают размер 1500 байт).

В каждом из SLIP-кадров полностью воспроизводится IP-заголовок размером 20 байт (рис. 5.11). Из-за этого избыточность, возникающая при передаче длинных пакетов по протоколу SLIP, весьма велика. Существенна и избыточность, порождаемая самим асинхронным методом передачи на интерфейсе ПК-модем (минимум 20 % на дополнительные стартовый и стоповый биты на каждый байт). Но с этим ничего поделать нельзя, поскольку все персональные компьютеры имеют только асинхронные порты.

Для установления связи по протоколу SLIP компьютеры должны иметь информацию об IP-адресах друг друга. В протоколе SLIP нет механизмов, обеспечивающих возможность обмениваться адресной информацией, так как в структуре кадра не предусмотрено поле адреса и его специальная обработка. Поэтому компьютерам, взаимодействующим по протоколу SLIP, должны быть назначены IP-адреса заранее. Каждый раз после установления SLIP-соединения компьютер превращается в полноправный хост Internet со своим собственным IP-адресом. Если провайдер использует динамическое присвоение IP-адресов, то при каждом новом соединении компьютер будет получать новый IP-адрес. Следовательно, другие компьютеры в сети будут вынуждены искать его под неизвестно каким адресом.

Другим недостатком протокола SLIP является отсутствие в нем индикации типа протокола, пакет которого инкапсулируется в SLIP-кадр. Поэтому через последовательную линию по протоколу SLIP можно передавать трафик лишь одного сетевого протокола. SLIP не позволяет различать пакеты по типу протокола, например, IP или DECnet. При работе по протоколу SLIP предполагается использование только протокола IP, что определено его названием Serial Line IP.

При работе с реальными телефонными линиями, зашумленными и поэтому искажающими информацию при пересылке, необходимы процедуры обнаружения и коррекции ошибок. В протоколе SLIP такие процедуры не предусмотрены. Эти функции обеспечивают вышележащие протоколы: протокол IP проводит тестирование целостности пакета по заголовку IP, а один из двух транспортных протоколов (UDP или TCP) проверяет целостность всех данных по контрольным суммам.

В стандартном SLIP не предусмотрено сжатие данных, но существуют его варианты со сжатием, например CSLIP. Большинство современных модемов, поддерживающих стандарты V.42bis и MNP5, осуществляют эту операцию аппаратно.

Низкая пропускная способность последовательных линий связи заставляет сокращать время передачи пакетов, уменьшая объем содержащейся в них служебной информации. Эта задача решается с помощью протокола Compressed SLIP (CSLIP), поддерживающего сжатие заголовков IP-пакетов.

Протокол CSLIP был создан в Lawrence Berkeley Labs (LBL) Ван Якобсоном как средство повышения эффективности последовательной передачи и уровня сервиса прикладных программ, использующих TCP/IP на медленных линиях. Протокол CSLIP, по сравнению с протоколом SLIP, использует в шесть раз меньше избыточной информации (в виде заголовков). На низких скоростях передачи данных эта разница заметна только при работе с IP-пакетами, несущими малые объемы информации, такие пакеты формируются, например, при работе telnet или rlogin. На больших же скоростях CSLIP дает меньший выигрыш и почти никакого выигрыша для пакетов с большими объемами данных, например ftp-пакетов.

Появление CSLIP объясняет тот факт, что при использовании программ типа telnet, rlogin и других для пересылки одного байта данных требуется переслать 40 байт служебной информации. При сжатии заголовков 20 октетов заголовка IP и 20 октетов заголовка TCP (итого 40 байт) заменяются 3 – 7 октетами. CSLIP для сжатия – распаковки и проверки правильности пересылки пакета (и заголовка) использует информацию из предыдущего пакета, т.е. передача имеет структуру цепочки. Первый пакет в цепочке – несжатый. Если какой-либо пакет теряется, то цепочка рвется, нельзя этот же пакет запросить в самом конце передачи, его нужно пересылать заново тут же, т.е. прекращать процесс передачи и начинать новую цепочку. Таким образом, эта технология при пропаже или искажении пакетов приводит к бóльшим потерям времени, чем обычный SLIP. Это происходит из-за задержек на останов и передачу нового несжатого пакета.

Так как в протоколе SLIP процедуры обнаружения и коррекции ошибок не предусмотрены, то нежелательно совместное использование дейтаграммного протокола UDP и SLIP. Это объясняется тем, что в протоколе UDP не обязательно применение контрольных сумм.

Дальнейшим развитием протокола SLIP является протокол PPP (RFC 1331), в котором устранены некоторые недостатки протокола SLIP. Необходимо помнить что SLIP и PPP – протоколы канального уровня.

Протокол PPP

Протокол Point-to-Point Protocol (PPP) (протокол канала связи с непосредственным соединением) был официально опубликован в 1993 г. и стал стандартом для связи по последовательным каналам, например таким, которые при-

меняют для обмена информацией между домашними компьютерами и Internet по коммутируемым телефонным линиям. PPP имеет значительные преимущества перед SLIP: компьютеры на противоположных концах соединения могут договариваться о параметрах сеанса связи и сообщать друг другу свои IP-адреса, которые в отличие от SLIP, могут назначаться динамически.

Кроме формирования стандартных IP-пакетов данных, протокол PPP призван решать и другие проблемы, в том числе:

- присвоение и управление IP-адресами;
- асинхронное и синхронное бит-ориентированное формирование пакета данных;
- мультиплексирование протокола сети;
- конфигурация канала связи;
- проверка качества канала связи;
- обнаружение ошибок;
- согласование адреса сетевого уровня;
- согласование протокола сжатия информации.

Протокол PPP решает эти задачи путем обеспечения расширяемого протокола управления каналом (LCP – Link Control Protocol) и семейства протоколов управления сетью (NCP – Network Control Protocols), которые позволяют согласовывать факультативные параметры конфигурации и различные возможности.

PPP реализует метод передачи дейтаграмм через последовательные каналы связи с непосредственным соединением. Протокол PPP содержит три основных компонента:

протокол управления каналом передачи данных высокого уровня (HDLC), используемый в качестве базиса для формирования дейтаграмм при прохождении через каналы с непосредственным соединением;

расширяемый протокол LCP – для организации, выбора конфигурации и проверки соединения канала передачи данных;

семейство протоколов NCP – для организации и выбора конфигурации различных протоколов сетевого уровня.

PPP обеспечивает одновременное пользование множеством протоколов сетевого уровня.

Основные принципы работы. Чтобы организовать связь через канал связи с непосредственным соединением, инициирующий PPP сначала отправляет пакеты LCP для выбора конфигурации и (факультативно) проверки канала передачи данных. После того, как канал установлен и пакетом LCP проведено необходимое согласование факультативных средств, инициирующий PPP отправляет пакеты NCP, чтобы выбрать и определить конфигурацию одного или более протоколов сетевого уровня. Как только конфигурация каждого выбранного протокола определена, дейтаграммы из каждого протокола сетевого уровня можно отправлять через данный канал. Канал сохраняет свою конфигурацию для связи до тех пор, пока явно выраженные пакеты LCP или NCP не

закроют этот канал или пока не произойдет какое-нибудь внешнее событие (например, истечет срок бездействия таймера или вмешается какой-нибудь пользователь).

Требования к интерфейсам физического уровня. PPP может работать через любой интерфейс DTE/DCE (например, EIA RS-232-C, EIA RS-422, EIA RS-423 и CCITT V.35). Единственным требованием, которое предъявляет PPP, является обеспечение дублированных схем (либо специально назначенных, либо переключаемых), которые могут работать как в синхронном, так и в асинхронном последовательном по битам режиме, прозрачном для блоков данных канального уровня PPP. Протокол не предъявляет каких-либо ограничений, касающихся скорости передачи информации, кроме тех, которые определены конкретным примененным интерфейсом DTE/DCE.

Канальный уровень PPP. PPP использует принципы, терминологию и структуру блока данных процедур HDLC (ISO 3309-1979), модифицированных стандартом ISO 3309-1984/PDAD1 (Приложение 1: Стартстопная передача). ISO 3309-1979 определяет структуру блока данных HLDC для применения в синхронных режимах передачи. ISO 3309-1984/PDAD1 определяет предложенные для стандарта ISO 3309-1979 модификации, которые позволяют его использование в асинхронных режимах. Формат блока данных PPP приведен на рис. 5.12.

Длина поля Flag составляет 1 байт; она указывает на начало или конец блока данных. Эта последовательность состоит из бинарной последовательности 01111110 – $(7E)_h$.

Длина поля *A*ddress тоже равна 1 байт; оно содержит бинарную последовательность 11111111 – $(FF)_h$, представляющую собой стандартный широковещательный адрес. PPP не присваивает индивидуальных адресов станциям.

Поле Control составляет 1 байт; содержит информацию о предоставляемых услугах.

Длина поля Protocol равна 2 байт; его значение идентифицирует протокол, заключенный в информационном поле блока данных. Значения поля Protocol и соответствующие им пакеты представлены в табл. 5.6.

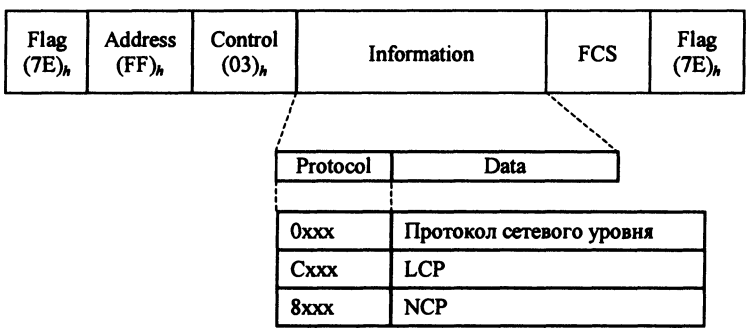


Рис. 5.12. Формат кадра PPP

Таблица 5.6. **Типы пакетов поля *Protocol***

Значение поля Protocol	Тип пакета
0021	IP
0023	ISO CLNP
0025	Xerox NS IDP
0027	DECnet Phase IV
0029	Apple Talk
002B	IPX
002D	Van Jacobson Compressed TCP/IP 1
002F	Van Jacobson Compressed TCP/IP 2
8021	IP Control Protocol
8023	ISO CLNP Control Protocol
8025	Xerox NS IDP Control Protocol
8027	DECnet Phase IV Control Protocol
8029	Apple Talk Control Protocol
802B	IPX Control Protocol
C021	Link Control Protocol
C023	User/Password Authentication Protocol

Поле Data содержит дейтаграмму для протокола, заданного в поле протокола. Конец информационного поля определяется замыкающей последовательностью Flag, перед которой размещается два байта поля FCS. Максимальная длина информационного поля по умолчанию равна 1500 байт. В соответствии с априорным соглашением, разрешающие реализации PPP могут использовать другие значения максимальной длины информационного поля.

Поле проверочной последовательности блока данных FCS (Frame Check Sequence) обычно составляет 16 бит (2 байт). В соответствии с априорным соглашением, разрешающие реализации PPP для усиления степени защиты от ошибок могут использовать 32-битовое (4-байтовое) поле FCS.

Протокол PPP инициируется, используя оба протокола LCP и NCP. Для PPP-соединения определено пять фаз:

- *организация канала* (Dead Phase). Эта фаза определяет физическую готовность канала прежде, чем может быть произведен обмен каких-либо дейтаграмм сетевого уровня (например IP). В случае успешной инициализации физического уровня канал переходит в следующую фазу;

• *определение качества канала связи и согласование его конфигурации (Establish Phase)*. Эта фаза инициализирует протокол LCP и определяет параметры канала. Здесь проверяется канал, чтобы определить, является ли качество канала достаточным для вызова протоколов сетевого уровня. Эта фаза завершается после того, как пакет подтверждения конфигурации (Configure ACK) будет принят на обоих концах канала. После этого канал считается открытым и переходит в фазу аутентификации (необязательно);

• *аутентификация (Authenticate Phase)*. В этой фазе аутентифицируются обе точки, используя протоколы PAP (Password Authentication Protocol) или CHAP (Challenge Handshake Authentication Protocol). Канал не переходит в следующую фазу Network Phase до успешной аутентификации. Если же аутентификация неуспешна, то канал переходит в фазу прекращения действия канала Terminate Phase;

• *согласование конфигурации протоколов сетевого уровня (Network Phase)*. После того как LCP завершит фазу определения качества канала связи, конфигурация сетевых протоколов может быть по отдельности выбрана соответствующими NCP. Сетевые протоколы могут быть в любой момент вызваны и освобождены для последующего использования. Если LCP закрывает данный канал, он информирует об этом протоколы сетевого уровня, чтобы они приняли соответствующие меры;

• *прекращение действия канала (Terminate Phase)*. Эта фаза закрывает PPP-канал административным путем. Иногда это происходит и из-за какого-нибудь физического события, такого, как плохое качество линии, потеря носителя или истечение периода бездействия таймера. Протокол LCP использует Terminate Request пакет для закрытия канала и сообщает соответствующим NCP, что канал закрыт.

Протокол управления канала связи LCP (Link Control Protocol). LCP обеспечивает метод организации, выбора конфигурации, поддержания и окончания работы канала с непосредственным соединением.

Существует три класса пакетов LCP:

для организации канала связи. Используются для организации и выбора конфигурации канала.

для завершения действия канала. Применяются для завершения действия канала связи.

для поддержания работоспособности канала. Используются для поддержания и отладки канала. К таким пакетам относятся, например, пакеты LQR (Link Quality Report) и Echo Request/Reply, обеспечивающие мониторинг качества стандартного синхронного канала LQM (Link Quality Monitoring).

LCP пакет передается в поле Information PPP-кадра. Формат пакета связи протокола LCP представлен на рис. 5.13.

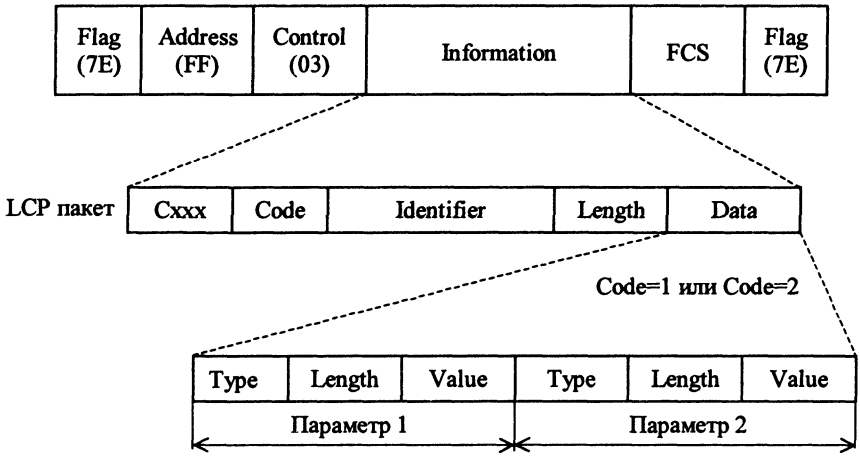


Рис. 5.13. Формат пакета LCP

Поле Code определяет тип LCP-сообщения, содержащегося в поле данных. Ниже приведены некоторые общие примеры LCP пакетов для организации канала.

1. Configure Request (Code = 1):
   а) открытие соединения;
   б) обмен параметрами конфигурации;
   в) прием оговоренных параметров от другой стороны.
2. Configure Ack (Code = 2):
   а) ответ на Configure Request;
   б) указание на то, что значения параметров, полученных в Configure Request, корректны;
   в) сигнализирует, что канал открыт по прибытию пакета.
3. Configure Nak (Code = 3) показывает, что значения параметров неприемлемы.
4. Configure Reject (Code = 4):
   а) указывает, что некоторые из параметров неприемлемы;
   б) шлет новый Configure Request пакет без неправильных параметров, найденных в отвергнутом пакете.

Поле Identifier определяет запросы и ответы. Поле Length указывает размер LCP-пакета. Поле Data содержит конфигурационные параметры для пакетов типа Configure Request и Configure Ack , определяемые полями Type, Length и Value. Всего может быть задано восемь параметров конфигурации. Поле Type описывает параметры конфигурации для согласования, поле Length описывает их длину, а поле Value определяет значение параметров конфигурации для согласования. В табл. 5.7 представлены параметры конфигурации LCP.

Таблица 5.7. **Параметры конфигурации LCP**

Параметр	Описание	Тип	Длина, байт	Значение
Maximum Receive Unit	Согласует размеры пакетов одного направления	1	4	1500 (по умолчанию)
Async-Control Character-Map	Согласует использование управляющих символов для асинхронных линий	2	6	FFFFFFFF (по умолчанию)
Authentication-Protocol	Используется для аутентификации, до того как начнется обмен пакетами третьего уровня	3	≥ 4	C023 (PAP) C223 (CHAP)
Quality-Protocol	Определяет мониторинг качества канала	4	≥ 4	Нет (по умолчанию) C025 (LQR)
Magic-Number	Определяет закольцованность канала и другие аномалии на втором уровне	5	6	Нет (по умолчанию)
RESERVED	—	6	—	—
Protocol-Field-Compressed	Согласует протокол сжатия на втором уровне	7	2	Запрещено (по умолчанию)
Address-and-Control Field-Compressed	Согласует сжатие полей Address и Control PPP-кадра	8	2	Не сжимать (по умолчанию)
FCS-Alternatives	Согласует 32-бит FCS	9	2	16 бит FCS (по умолчанию)

Семейство протоколов управления сетью NCP(Network Control Protocols). После фазы LCP в PPP-канале следует фаза согласования параметров, специфичных для каждого из протоколов сетевого уровня в NCP. Например, протокол IP требует от обоих концов обмена и приема соответствующих IP-адресов. NCP определяет, каким образом данные различных протоколов расположены внутри PPP-кадра, что описано отдельным соглашением RFC, определяющим как данный протокол инкапсулируется в PPP-кадр. Каждый NCP устанавливает, обслуживает и закрывает использование определенного протокола. На рис. 5.14 показан пример NCP кодов, которые могут находиться в поле Protocol PPP-кадра.

Аутентификация для PPP описана в RFC 1334. Она использует два протокола PAP и CHAP. Последний наиболее распространен в современных сетях. CHAP-аутентификация является частью фазы установления соединения LCP и повторяется через определенные интервалы времени, когда PPP-канал установлен.

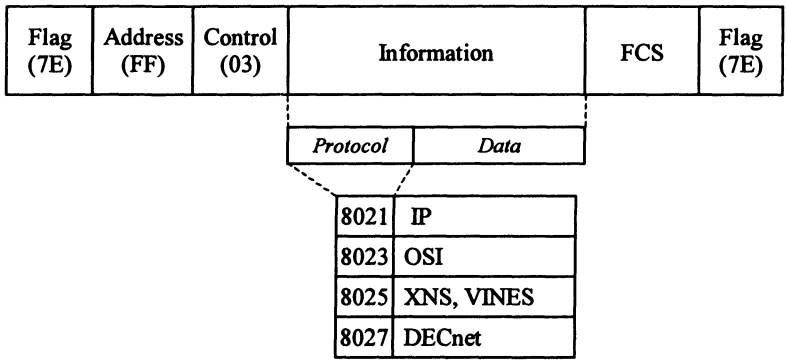


Рис. 5.14. Пример *NCP*-кодов

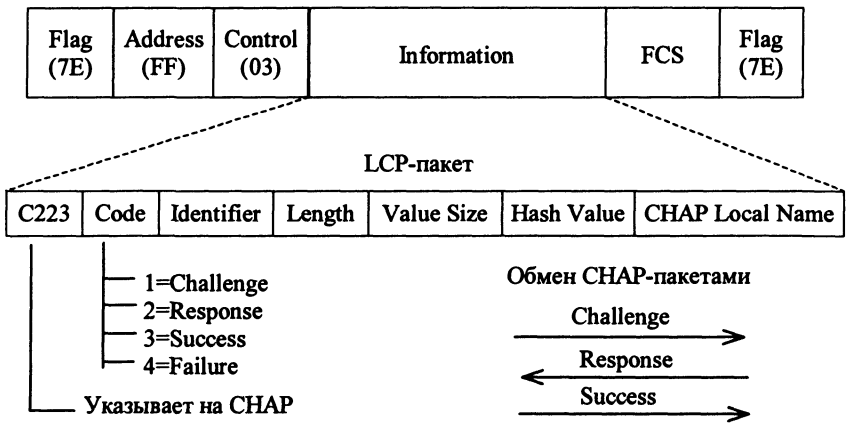


Рис. 5.15. CHAP-аутентификация

CHAP выполняется на каждой стороне канала и каждой стороне назначается одинаковый открытый ключ – secret, на базе которого вычисляется уникальное значение Hash Value, помещаемое в пакет (рис. 5.15). Аутентификация реализуется по алгоритму с открытым ключом.

Мониторинг качества канала LQM (*Link Quality Monitoring*). Мониторинг поддерживается только на стандартных синхронных каналах. Качество канала определяется процентом успешно переданных или полученных пакетов в пределах отчетного периода. Эти пакеты называются LQR (Link Quality Report). Процедура LQM использует LQR-пакеты, содержащие счетчики для определения входного и выходного качества канала для входных и выходных пакетов. Отчетный период описывается в конфигурации. После пяти отчетных периодов LQM вычисляет среднее значение процента переданных и принятых пакетов и сравнивает его с пороговым значением. Если средний процент меньше, чем пороговое значение, то соответствующий NCP закрывается. В допол-

нение к посылке LQR пакетов маршрутизатор может быть сконфигурирован для периодической посылки по каналу пакетов Echo Request. Если маршрутизатор послал определенное их количество и не получил ни одного пакета Echo Reply, то все NCP закрываются. Интервал между пакетами Echo Request и максимальное число пакетов оставшихся без ответа являются необязательными параметрами.

5.5. Протоколы III уровня стека TCP/IP

Протокол IPv4

Протокол межсетевого взаимодействия IP (Internet Protocol) – протокол ненадежной доставки. Ненадежность возникает только тогда, когда не хватает ресурсов или происходят сбои в используемых физических сетях.

Протокол IP определяет базовый элемент передачи данных, используемый во всем стеке TCP/IP. Программное обеспечение IP выполняет функцию маршрутизации, выбора пути, по которому будут передаваться данные. Помимо точной, формальной спецификации форматов данных и функции маршрутизации, IP включает набор правил, которые обеспечивают ненадежную доставку пакетов. Эти правила указывают, как маршрутизаторам следует обрабатывать пакеты, как и когда следует генерировать сообщения об ошибках, и условия, при которых можно удалять пакеты.

Протокольный блок данных IP называется межсетевой дейтаграммой (IP-дейтаграммой или просто дейтаграммой). Как и кадр канального уровня, дейтаграмма делится на поле заголовка и поле данных, заголовок дейтаграммы содержит адреса отправителя, получателя и поле типа, которое идентифицирует содержимое дейтаграммы. Разница между ними состоит в том, что заголовок дейтаграммы содержит IP-адреса, а заголовок кадра – физические адреса. Формат дейтаграммы представлен на рис. 5.16.

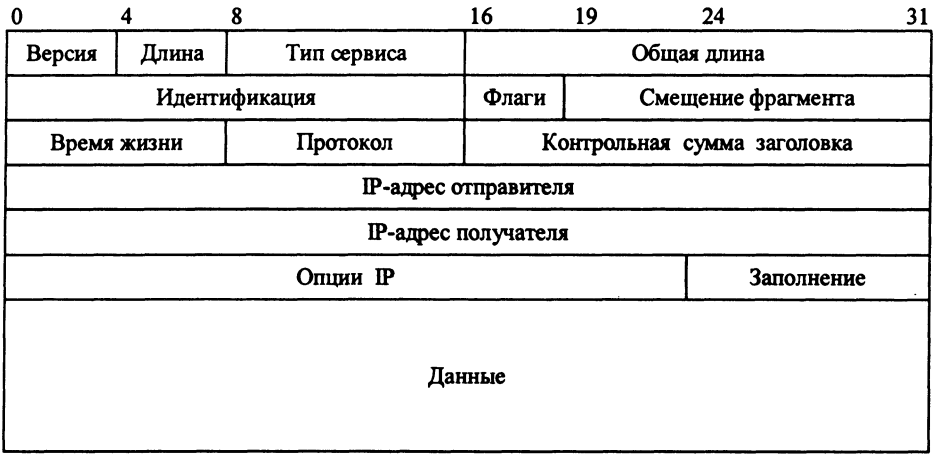


Рис. 5.16. Формат дейтаграммы

Так как обработка дейтаграммы происходит с помощью программного обеспечения, оборудование не накладывает никаких ограничений на ее содержимое и формат. Например, первое 4-битовое поле «Версия» в дейтаграмме содержит версию протокола IP, используемую при создании дейтаграммы. Оно используется отправителем, получателем, и всеми маршрутизаторами между ними для уверенности подтверждения того, что все они используют один и тот же формат дейтаграммы. Всему программному обеспечению IP необходимо проверять поле версии перед обработкой дейтаграммы для подтверждения того, что ее формат соответствует формату, который ожидает это обеспечение. Если стандарт меняется, машины будут отбрасывать дейтаграммы с версией протокола, отличающейся от версии, на которой они работают, предохраняя себя от неправильной интерпретации содержимого дейтаграммы из-за устаревшего формата. В настоящее время наибольшее распространение получила версия IPv4, которую постепенно сменит более совершенная IPv6. Рассмотрим последовательно эти версии.

Поле длины заголовка «Длина» также занимает 4 бита и хранит длину заголовка дейтаграммы в 32-битных словах. Все поля в заголовке имеют фиксированную длину, за исключением поля «Опции IP» и соответствующего ему поля «Заполнение». Наиболее простой заголовок, без опций и заполнения, занимает 20 октетов и имеет в поле заголовка «Длина» значение 5.

Поле «Общая длина» определяет длину IP-дейтаграммы, измеренную в октетах, включая октеты в заголовке и данных. Размер области данных можно вычислить путем вычитания длины заголовка «Длина» из «Общей длины». Так как поле «Общая длина» занимает 16 бит, то максимально возможный размер дейтаграммы IPv4 составляет 65535 октет.

Поле «Тип сервиса» 8 бит указывает, как следует обрабатывать дейтаграмму. Это поле разделено на пять подполей (рис. 5.17). Три бита «Приоритета» указывают приоритет дейтаграммы, значения которого могут меняться от 0 (обычный приоритет) до 7 (управление сетью), позволяя отправителям передавать информацию о важности каждой дейтаграммы, например управляющая информация может иметь больший приоритет, чем данные.

Биты D, T и R описывают тип передачи, который нужен дейтаграмме. Установка бита D запрашивает минимальные задержки при передаче, бита T – максимальную пропускную способность, а бита R – максимальную надежность. Конечно, межсетевое взаимодействие не может гарантировать выполнение запрошенного сервиса (например, может быть так, что нет пути к назначению с запрошенными качествами). Поэтому можно рассматривать запрос сервиса как указание алгоритмам маршрутизации, а не как требование. Если маршрутизатор знает более чем один маршрут к указанному назначению, он может

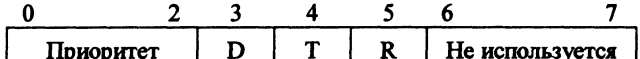


Рис. 5.17. Формат поля «Тип сервиса» IPv4

использовать поле типа передачи для выбора пути с характеристиками наиболее близкими требуемым. Например, предположим, что маршрутизатор может выбирать между низкоскоростной арендованной линией или высокоскоростной (но с большими паузами) спутниковой линией. Дейтаграммы, передающие нажатия клавиш от пользователя к удаленному компьютеру, могут иметь установленным бит D, запрашивая самую быструю доставку, в то время как дейтаграммы, используемые при передаче большого файла, могут иметь установленным бит T, запрашивая передачу по высокоскоростной спутниковой линии.

Важно понимать, что алгоритмы маршрутизации должны выбирать одну из возможных используемых физических сетевых технологий, которые имеют определенные характеристики задержки, пропускной способности и надежности. Часто данная технология использует компромисс между этими характеристиками (например, высокой скоростью передачи и большей задержкой). Поэтому, идея состоит в том, чтобы дать алгоритму указание о том, что наиболее важно; не имеет смысла указывать все три типа сервиса.

Таким образом, спецификацию типа передачи следует рассматривать как указание алгоритму маршрутизации, которое помогает выбрать один путь к сети назначения из возможных на основе знаний об аппаратных технологиях, использующихся на этих путях. Интернет не гарантирует выполнения запрошенного транспортного сервиса.

Инкапсуляция дейтаграмм. В отличие от кадров физической сети, которые должны распознавать оборудование, дейтаграммы обрабатываются программным обеспечением. Их длина не должна превышать 65535 октетов.

Более жесткие ограничения на размер дейтаграммы накладывают параметры сетей. При передаче дейтаграммы от одной машины к другой, ее должны транспортировать базовые физические сети. Для эффективности межсетевой передачи нужно быть уверенным в том, что каждая дейтаграмма передается в отдельном физическом кадре. Передача одной дейтаграммы в одном кадре называется инкапсуляцией. Для базовой сети дейтаграмма выглядит так же, как и любое сообщение, посылаемое от одной машины к другой. Оборудование как не знает формата дейтаграммы, так и не понимает IP-адреса назначения. Поэтому, когда одна машина посылает IP-дейтаграмму другой, вся дейтаграмма передается как часть данных кадра (рис. 5.18).

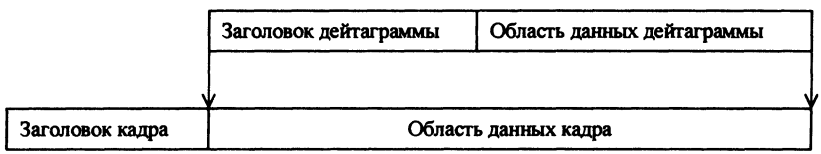


Рис. 5.18. Инкапсуляция IP-дейтаграммы в кадр

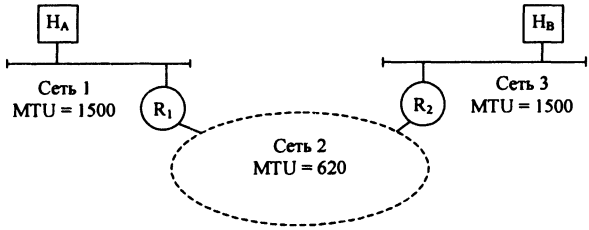


Рис. 5.19. Пример применения фрагментации

В идеальном случае вся IP-дейтаграмма помещается в одном физическом кадре, что делает передачу по физической сети более эффективной. Каждая технология коммутации пакетов устанавливает фиксированную верхнюю границу количества данных, которые могут быть переданы в одном физическом кадре. Такие ограничения называют «максимальная единица передачи» – MTU (Maximum Transfer Unit). Значение MTU может быть довольно маленьким: некоторые аппаратные технологии ограничивают размер передачи до 128 октет или даже меньше. Маленькие части, на которые делится дейтаграмма, называются *фрагментами*, а процесс деления дейтаграммы – *фрагментацией*.

Как показывает рис. 5.19, фрагментация обычно осуществляется маршрутизатором где-либо на пути между отправителем дейтаграммы и ее истинным получателем. Маршрутизатор принимает дейтаграмму из сети с большим MTU и должен передать ее по сети, в которой MTU меньше, чем размер дейтаграммы. На рисунке оба компьютера H_A и H_B напрямую присоединены к сетям Ethernet (Сеть 1 и Сеть 3), которые имеют MTU в 1500 октет. Поэтому оба компьютера могут генерировать и посылать дейтаграммы до 1500 октет длины. Путь между ними, тем не менее, включает сеть с MTU = 620 октет. Если H_A посылает H_B дейтаграмму длиннее 620 октет, маршрутизатор R_1 будет фрагментировать эту дейтаграмму. Аналогично, если H_B посылает большую дейтаграмму H_A, маршрутизатор R_2 будет фрагментировать эту дейтаграмму. Размер фрагмента выбирается таким, чтобы каждый фрагмент мог транспортироваться в одном кадре. Кроме того, так как IP передает значение смещения данных в восьмерках октетов, размер фрагмента должен быть выбран кратным восьми. Конечно, выбор числа октетов, кратного восьми и наиболее близкого к сетевой MTU, не обязательно делит дейтаграмму на равные части; последняя часть часто короче, чем остальные. Фрагменты должны собираться для воссоздания полной копии исходной дейтаграммы перед тем, как она будет обрабатываться у получателя.

Протокол IP как не ограничивает дейтаграммы до маленького размера, так и не гарантирует, что большие дейтаграммы будут доставлены без фрагментации. Отправитель может выбрать любой размер дейтаграммы. Фрагментация и сборка производятся автоматически, не требуя специальных действий от отправителя. Спецификация IP устанавливает, что маршрутизаторы должны

Рис. 5.20. Исходная дейтаграмма, несущая 1400 октетов данных (*а*) и три фрагмента для сети с MTU= 620 (*б*)

принимать дейтаграммы с размерами, не превосходящими MTU сетей, к которым они присоединены. Кроме того, маршрутизаторы должны всегда обрабатывать дейтаграммы размером до 576 октет.

Фрагментация дейтаграммы означает разделение ее на несколько частей. Рис. 5.20 иллюстрирует результат фрагментации.

Каждый фрагмент содержит заголовок дейтаграммы, дублирующий большую часть заголовка исходной дейтаграммы (кроме бита в поле «Флаги», который показывает, что это фрагмент), и столько данных, сколько может содержать фрагмент, чтобы общая длина была меньше, чем MTU сети, по которой он путешествует.

Сборка фрагментов. В сетях TCP/IP после фрагментирования дейтаграммы, ее фрагменты передаются как отдельные дейтаграммы на всем протяжении пути до места назначения, где они собираются. Сохранение фрагментации на всем протяжении пути имеет два недостатка. Во-первых, так как дейтаграммы не собираются сразу же после прохождения сети с маленьким MTU, маленькие фрагменты будут передаваться с места фрагментации до места назначения. Сборка дейтаграмм в месте назначения может привести к неэффективности: даже если некоторые сети, проходимые после фрагментации, имеют большое значение MTU, их будут пересекать только маленькие фрагменты. Во-вторых, если какой-либо фрагмент будет потерян, дейтаграмму нельзя будет восстановить. Принимающая машина запускает таймер сборки при приеме первого фрагмента. Если таймер обнуляется до того, как приняты все фрагменты, принимающая машина удаляет оставшиеся части, не обрабатывая дейтаграмму. Поэтому вероятность потери дейтаграммы увеличивается при использовании фрагментации, так как потеря одного фрагмента приводит к потере всей дейтаграммы.

Несмотря на эти недостатки, выполнение сборки в месте назначения хорошо работает. Оно позволяет независимо маршрутизировать фрагменты и не требует от промежуточных маршрутизаторов хранения собираемых фрагментов.

Управление фрагментацией. Три поля в заголовке дейтаграммы – Идентификация, Флаги и Смещение фрагмента – управляют фрагментацией и сборкой дейтаграмм (см. рис. 5.16). Поле «Идентификация» содержит уникальное целое число, которое идентифицирует дейтаграмму. Напомним, что когда шлюз или маршрутизатор фрагментирует дейтаграмму, он копирует бóльшую часть полей в заголовке дейтаграммы в каждый фрагмент. Поле «Идентификация» позволяет получателю узнать, какой дейтаграмме принадлежат прибывающие фрагменты. Когда появляется фрагмент, получатель для идентификации дейтаграммы использует поле «Идентификация» вместе с полем адреса источника. Компьютер, посылающий IP-дейтаграммы, должен генерировать уникальное значение поля «Идентификация» для каждой отдельной дейтаграммы (в теории, повторные передачи дейтаграммы должны содержать то же самое значение в поле «Идентификация», что и в исходной дейтаграмме; на практике, протоколы высокого уровня обычно выполняют повторную передачу как новую дейтаграмму со своим значением поля «Идентификация»). Один из модулей, используемых в программном обеспечении IP, хранит глобальный счетчик в памяти, инкрементирует его каждый раз, когда создается новая дейтаграмма, и копирует результат в поле «Идентификация» дейтаграммы.

Напомним, что каждый фрагмент имеет точно такой же формат, что и полная дейтаграмма. Для фрагмента поле «Смещение фрагмента» указывает смещение в исходной дейтаграмме данных, передаваемых в фрагменте, измеряемое в 8 октетах (смещения измеряются в восьмерках октетов для сохранения места в заголовке), начиная со смещения ноль. Для сборки дейтаграммы это поле должно получить назначение во всех фрагментах, начиная с фрагмента со смещением 0 до фрагмента с наибольшим смещением. Фрагменты необязательно прибывают по порядку, и не существует прямого взаимодействия между маршрутизатором, который фрагментирует дейтаграммы, и получателем, который пытается собирать их.

Младшие два бита из трехбитового поля «Флаги» управляют фрагментацией. Обычно прикладное программное обеспечение, использующее TCP/IP, не заботится о фрагментации, так как и фрагментация, и сборка являются автоматическими процедурами, выполняемыми на низком уровне в операционной системе незаметно для пользователя. Тем не менее, для тестирования межсетевого программного обеспечения или отладки рабочих программ может оказаться важной проверка размеров дейтаграмм, для которых осуществляется фрагментация. Первый управляющий бит помогает при таком тестировании, указывая возможность фрагментации дейтаграммы. Он называется битом «не фрагментировать», так как установка его в единицу указывает, что дейтаграмму нельзя фрагментировать. Приложение может выбрать запрет фрагментации, когда нужна лишь целая дейтаграмма. Всякий раз, когда маршрутизатору нужно фрагментировать дейтаграмму с установленным битом «не фрагментировать», он удаляет дейтаграмму и посылает обратно источнику сообщение об ошибке.

Младший бит в поле «Флаги» указывает, содержит ли фрагмент данные из середины дейтаграммы или из конца. Он называется битом «еще фрагменты». Поясним необходимость наличия этого бита. Программное обеспечение *IP* у получателя будет получать фрагменты (возможно не по порядку), и ему нужно будет знать, когда оно получит все фрагменты дейтаграммы. Когда поступает очередной фрагмент, поле «Общая длина» в заголовке указывает размер фрагмента, а не размер всей дейтаграммы, поэтому получатель не может использовать поле «Общая длина» для того, чтобы определить, собрало ли он все фрагменты. Бит «еще фрагменты» легко решает проблему: как только получатель получает фрагмент со сброшенным битом «еще фрагменты», он знает, что этот фрагмент несет в себе данные из конца исходной дейтаграммы. На основе полей «Смещение фрагмента» и «Общая длина» оно может вычислить длину исходной дейтаграммы. Проверив «Смещение фрагмента» и «Общая длина» у всех прибывших фрагментов, получатель может определить, содержат ли фрагменты все данные, требуемые для сборки исходной дейтаграммы.

Время жизни дейтаграммы. Поле «Время жизни» указывает, сколько секунд может оставаться дейтаграмма в межсетевой системе. Эта идея является насколько простой, настолько и важной: всякий раз, когда машина передает дейтаграмму в Интернет, она устанавливает максимальное время существования дейтаграммы. Шлюзы и маршрутизаторы, обрабатывающие дейтаграмму, должны декрементировать поле «Время жизни» по мере того, как идет время, и удалять дейтаграмму из Интернета, когда время вышло.

Оценить время точно трудно, так как маршрутизаторы обычно не знают время передачи между физическими сетями. Принятые соглашения упрощают обработку и делают легкой обработку дейтаграмм без синхронизации часов. Во-первых, каждому маршрутизатору на пути от источника к назначению требуется декрементировать поле «Время жизни» на единицу, когда он обрабатывает заголовок дейтаграммы. В случае, когда маршрутизаторы перегружены, что приводит к большим паузам при передаче, каждый маршрутизатор хранит локальное время прихода дейтаграммы и декрементирует «Время жизни» на число секунд, в течение которого дейтаграмма находилась в маршрутизаторе, ожидая обслуживания.

Всякий раз, когда поле «Время жизни» обнуляется, маршрутизатор удаляет дейтаграмму и посылает сообщение об ошибке обратно источнику. Поле «Время жизни» гарантирует, что дейтаграммы не смогут вечно путешествовать по Интернету, даже если таблицы маршрутизации разрушатся и маршрутизаторы будут маршрутизировать дейтаграммы по кольцу.

Значение в поле «Протокол» указывает, какой протокол высокого уровня использовался при создании сообщения, передаваемого в области «Данные» дейтаграммы. По существу, значение в «Протокол» специфицирует формат области «Данные». Соответствие между протоколом высокого уровня и целым числом, используемым в поле «Протокол», должно устанавливаться ответственным центром, чтобы гарантировать действие соглашения по всему Интернету.

Поле «Контрольная сумма заголовка» удостоверяет целостность значений полей заголовка. Контрольная сумма IP формируется путем представления заголовка как последовательности 16-битовых чисел (с сетевым порядком байт), сложения их вместе, используя арифметику с дополнительным представлением отрицательных чисел, и получения отрицания числа. При вычислении контрольной суммы поле «Контрольная сумма заголовка» предполагается равным нулю. Необходимо помнить, что эта контрольная сумма применима только к числам, находящимся в заголовке IP, а не в данных. Разделение контрольной суммы для заголовка и для данных имеет свои преимущества и недостатки. Так как заголовок обычно занимает меньше октетов, чем данные, наличие отдельной контрольной суммы для него уменьшает время обработки в маршрутизаторах, которые вычисляют только контрольную сумму заголовка. Это разделение также позволяет протоколам более высокого уровня выбирать свои собственные схемы расчета контрольной суммы для данных. Главным недостатком является то, что протоколы более высокого уровня вынуждены добавлять свои контрольные суммы или рисковать тем, что они не смогут обнаружить искажения данных.

IP-адреса. Поля «Адрес отправителя IP» и «Адрес получателя IP» содержат 32-битовые IP-адреса отправителя и конечного получателя дейтаграммы IPv4. Хотя дейтаграмма может проходить через большое число промежуточных шлюзов или маршрутизаторов, поля отправителя и получателя никогда не изменяются; они указывают IP-адреса истинного источника и конечного назначения.

IP-адрес состоит из адреса сети и адреса узла в этой сети и принадлежит одному из пяти классов (рис. 5.21). Класс данного IP-адреса можно определить по первым трем старшим битам, причем первых двух бит достаточно для определения принадлежности адреса к одному из трех основных классов. Адреса класса A выделяют под адрес сети 7 бит, а под адрес узла 24 бит. Адреса класса B выделяют 14 бит под адрес сети и 16 бит под адрес узла. И наконец, сети класса C, состоящие менее чем из 256 узлов, выделяют 21 бит под адрес сети и только 8 бит под адрес узла. Маршрутизаторы используют для маршрутизации поле «Адреса сети». Характеристики классов IP-адресов представлены в табл. 5.8.

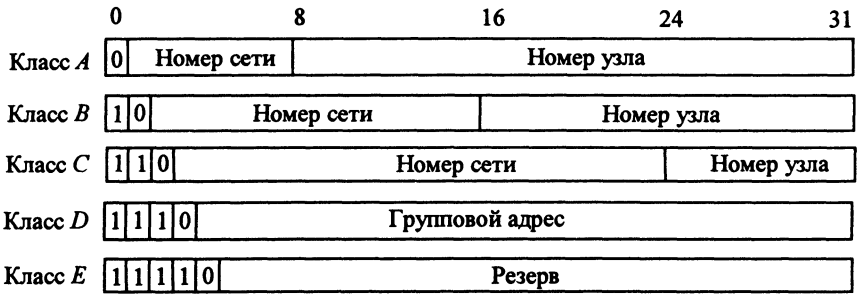


Рис. 5.21. Структура IP-адресов протокола IPv4

Таблица 5.8. **Классы IP-адресов**

Класс IP-адреса	Диапазон значений первого октета	Возможное количество сетей	Возможное количество узлов
A	1 – 126	126	16777214
B	128 – 191	16382	65534
C	192 – 223	2097150	254
D	224 – 239	–	2^{28}
E	240 – 247	–	2^{27}

Когда обычные компьютеры имеют два или более физических соединений, их называют многоадресными (multi-homed). Многоадресные узлы требуют нескольких адресов IP. Каждый адрес соответствует одному из соединений этой машины с сетью. Так как IP-адреса кодируют как сеть, так и узел в этой сети, они не описывают конкретную машину, а только соединение ее с сетью. Поэтому маршрутизатор, соединяющий N сетей, имеет N различных IP-адресов, по одному на каждое сетевое соединение.

Межсетевые адреса могут использоваться для указания как на сети, так и на отдельные узлы. По соглашению адрес сети имеет поле «адреса узла», равное нулю. IP-адреса могут использоваться для указания широковещания и отображения его в аппаратное широковещание. По соглашению, широковещательный адрес имеет поле адреса узла со всеми битами, равными «1». Такой широковещательный адрес называют направленным (directed) широковещательным адресом, так как он содержит как корректный идентификатор сети, так и корректный широковещательный адрес узла. Направленный широковещательный адрес может однозначно интерпретироваться в любой точке Интернета, так как он идентифицирует уникальным образом сеть получателя помимо указания на широковещание в этой сети. Направленные широковещательные адреса обеспечивают мощный (и чем-то опасный) механизм, позволяющий удаленной системе посылать один пакет, который будет распространен в режиме широковещания в указанной сети.

С точки зрения адресации, главным недостатком направленного широковещания является то, что оно требует знаний об адресе сети. Другая форма широковещательного адреса, называемая ограниченный широковещательный адрес или локальный сетевой широковещательный адрес, обеспечивает широковещательный адрес для локальной сети (сети отправителя), независимо от назначенного ей IP-адреса. Локальный широковещательный адрес состоит из 32 единиц (поэтому он иногда называется широковещательным адресом из всех единиц). Узел может использовать ограниченный широковещательный адрес в процессе своего запуска, до того, как он узнает свой IP-адрес или IP-адрес локальной сети. Как только узел узнает IP-адрес своей сети, он может использовать направленное широковещание.

Если поле адреса, состоящее из единиц, может интерпретироваться как «все узлы» в сети, то поля, состоящие из нулей, межсетевое программное обеспечение интерпретирует как символ «это». Поэтому IP-адрес с адресом узла, равным «0», обозначает «этот узел», а межсетевой адрес с идентификатором сети «0» обозначает «эта сеть». Конечно, использовать эти адреса нужно только в том контексте, в котором они интерпретируются однозначно.

Помимо широковещания схема адресов IP поддерживает специальную форму групповой доставки, известную как групповая доставка (multicasting). Групповая доставка особенно полезна для сетей, в которых аппаратная технология поддерживает групповую доставку.

При описании IP-адресов принято использовать точечную десятичную нотацию. IP-адреса записывают как четыре десятичных числа, разделенных десятичными точками, и каждое из этих чисел представляет значение одного октета IP-адреса. Например, IP-адрес 195.190.21.45 соответствует узлу 45 сети 195.190.21.0 класса *C*.

Сетевой адрес класса *A* 127.0.0.0 зарезервирован для обратной связи и введен для тестирования взаимодействия между процессами на одной машине. Когда какая-либо программа использует адрес обратной связи для передачи данных, протокольное программное обеспечение в компьютере возвращает эти данные, ничего не посылая по сети. Таким образом, пакет, посланный в сеть с адресом 127, не будет передаваться ни по какой сети. Более того, узел или маршрутизатор никогда не должен распространять информацию о маршрутах для сети с номером 127; этот адрес не является адресом сети.

Интерпретация выделенных адресов IPv4 представлена в табл. 5.9.

Таблица 5.9. **Интерпретация адресов IP**

Поле адреса		Интерпретация
Адрес сети	Адрес узла	
Все нули		Данный узел
Номер сети	Все нули	Данная IP-сеть
Все нули	Номер узла	Узел в данной (локальной) IP-сети
Все единицы		Все узлы в данной (локальной) IP-сети
Номер сети	Все единицы	Все узлы в указанной IP-сети
127	–	«Петля»

Поле «Данные», представленное на рис. 5.16, показывает начало области данных в дейтаграмме. Его длина зависит от того, что посылается в дейтаграмме.

Поле «Опции IP», рассматриваемое ниже, имеет переменную длину. Поле «Заполнение» зависит от выбранных опций. Оно представляет собой биты, содержащие нули, которые могут потребоваться для дополнения заголовка дейтаграммы до длины, кратной 32 бит (напомним, что поле длины заголовка указывает ее в 32-битных словах).

Межсетевые опции дейтаграммы. Поле «Опции IP», следующее за адресом назначения, не нужно каждой дейтаграмме; опции включаются, в основном,

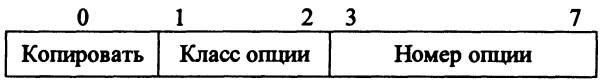


Рис. 5.22. Межсетевые опции IP-дейтаграммы

для тестирования или отладки сети. Обработка опций, тем не менее, является составной частью протокола IP, поэтому все стандартные реализации включают ее. Длина поля «Опции IP» зависит от выбранных опций. Когда в дейтаграмме есть опции, они размещаются друг за другом, без специальных разделителей между ними. Каждая опция состоит из кода опции длиной 1 октет, за которым может следовать длина опции (тоже занимает 1 октет) и группы октетов данных для этой опции. Октет кода опции делится на три поля: 1-битовый флаг «Копировать», 2-битовый «Класс опции» и 5-битовый «Номер опции» (рис. 5.22). Флаг «Копировать» управляет тем, как маршрутизаторы рассматривают опции при фрагментации. Когда бит «Копировать» установлен в 1, он указывает, что эта опция должна копироваться во все фрагменты. Когда он установлен в 0 – опцию нужно копировать только в первый фрагмент, а не во все.

Таблица 5.10. **Классы опций протокола Ipv4**

Класс опции	Значение
0	Управление дейтаграммой или сетью
1	Зарезервировано
2	Отладка и измерения
3	Зарезервировано

Биты «Класс опции» и «Номер опции» указывают общий класс опции и номер опции внутри этого класса. В табл. 5.10 представлено значение номера класса. В табл. 5.11 приведен список возможных опций в IP-дейтаграммах и указаны для них значения полей «Класс опции» и «Номер опции». Как видно из этого списка, бóльшая часть опций предназначена для управления.

Таблица 5.11. **Список опций протокола IPv4**

Класс опции	Номер опции	Длина	Описание
0	0	–	Конец списка опций. Используется, если опция не заканчивается в конце заголовка (смотри также поле дополнения)
0	1	–	Нет операции (используется для выравнивания октетов в списке опций)
0	2	11	Секретность (для военных приложений)
0	3	Переменная	Слабая маршрутизация источника. Используется для маршрутизации дейтаграммы по указанному пути
0	7	Переменная	Запись маршрута. Используется для трассировки маршрута
0	8	4	Идентификатор потока. Используется для передачи идентификатора потока SATNET (недействительно)
0	9	Переменная	Сильная маршрутизация источника. Используется для маршрутизации дейтаграммы по указанному пути
2	4	Переменная	Межсетевые временные метки. Используется для записи временных меток по маршруту

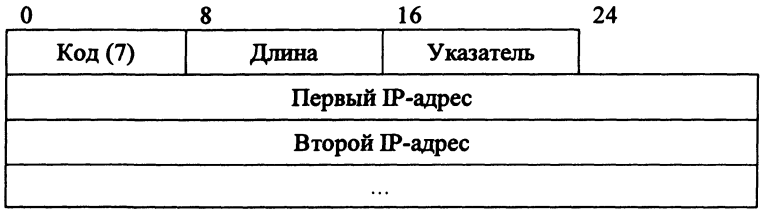


Рис. 5.23. Формат опции записи маршрута

Опция записи маршрута. Опции маршрутизации и временны́х меток являются самыми интересными, так как они обеспечивают способ наблюдения или управления тем, как маршрутизируются дейтаграммы. Опция запись маршрута позволяет источнику создать пустой список IP-адресов и заставить каждый маршрутизатор, обрабатывающий дейтаграмму, добавлять свой IP-адрес к этому списку. Рис. 5.23 иллюстрирует формат опции записи маршрута.

Как описано выше, поле «Код» содержит номер опции и класс опции (7 – для записи маршрута). Поле «Длина» указывает общую длину опции в том виде, в котором она представлена в IP-дейтаграмме, включая первые три октета. Поля, начиная с поля, помеченного «Первый IP-Адрес», составляют область, зарезервированную под хранение межсетевых адресов. Поле «Указатель» определяет смещение внутри опции первого свободного слота в списке.

Всякий раз, когда машина обрабатывает дейтаграмму, имеющую опцию записи маршрута, она добавляет свой адрес к списку записи маршрута (в опции должно быть выделено достаточно места исходным отправителем для того, чтобы поместились все нужные элементы). При добавлении своего адреса к списку машина сначала сравнивает поля указателя и длины. Если указатель больше, чем длина, то список полон, и машина отправляет дейтаграмму, не добавляя нового элемента. Если список не полон, машина вставляет 4-байтовый IP-адрес в позицию, определенную «Указателем», и увеличивает значение «Указателя» на четыре.

При прибытии дейтаграммы машина-получатель должна выделить и обработать список IP-адресов. Если получатель обрабатывает дейтаграмму обычным образом, он будет игнорировать записанный путь. Следует отметить, что отправитель должен разрешить наличие опции записи маршрута, а получатель должен быть согласен обработать полученный список; сама по себе машина не получит информацию о пройденном пути автоматически, если она включит опцию записи маршрута.

Опции пути источника. Идея, лежащая в основе маршрутизации источника, заключается в том, чтобы отправитель мог определять путь в Интернете. Например, для тестирования пропускной способности конкретной физической сети N-системные администраторы могут использовать маршрутизацию источника для направления IP-дейтаграмм через сеть N, даже если маршрутизаторы обычно выбирают путь, не включающий ее. Возможность делать такие тесты особенно важна в производственной среде, так как позволяет сетевым администраторам маршрутизировать дейтаграммы пользователей по сетям,

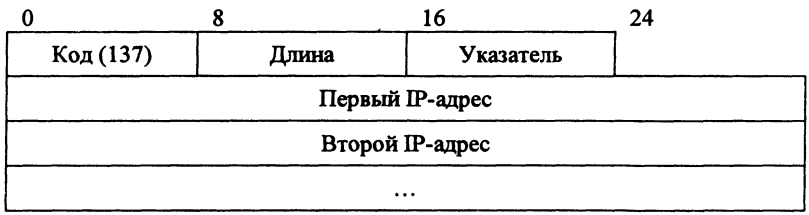


Рис. 5.24. Формат строгой маршрутизации

про которые известно, что они работают корректно, и параллельно с этим проверять другие сети. Конечно, такая маршрутизация полезна только для сетевых администраторов или квалифицированных пользователей, которые понимают топологию сети.

Протокол IP поддерживает две формы маршрутизации источника. Одна форма, названная строгой маршрутизацией источника, определяет путь с помощью включения последовательности IP-адресов в эту опцию (рис. 5.24). Строгая маршрутизация источника означает, что адреса определяют точный путь, которым должна следовать дейтаграмма при передаче ее к месту назначения. Путь между двумя последовательными адресами в списке должен состоять из одной физической сети; если маршрутизатор не может выполнить строгую маршрутизацию источника, возникает ошибка. Другая форма, называемая слабой маршрутизацией источника, также включает последовательность IP-адресов. Она определяет, что дейтаграмма должна следовать через эту последовательность IP-адресов, но допускает наличие нескольких переходов через сети между последовательными адресами в списке.

Обе опции маршрутизации источника требуют от маршрутизаторов на всем пути заменять элементы списка адресов своими сетевыми адресами. Поэтому, когда дейтаграмма поступает к получателю, она содержит список всех посещенных адресов, точно такой же, как и список, создаваемый опцией записи маршрута.

Формат опции маршрутизации источника напоминает показанный выше формат опции записи маршрута. Каждый маршрутизатор проверяет поля «Указатель» и «Длина», чтобы обнаружить переполнение списка. Если это произошло, указатель будет больше, чем длина, и маршрутизатор будет маршрутизировать дейтаграмму к ее назначению обычным образом. Если список заполнен еще не до конца, маршрутизатор на основании указателя выделяет IP-адрес, заменяет его на свой адрес (маршрутизатор имеет по одному адресу для каждого интерфейса; он записывает адрес, соответствующий сети, по которой он отправляет дейтаграмму) и маршрутизирует дейтаграмму, используя адрес, полученный из списка.

Опция временных меток. Эта опция работает аналогично опции записи маршрута в том отношении, что опция временных меток содержит вначале пустой список, а каждый шлюз на всем протяжении пути от источника к назначению заполняет элемент в этом списке. Каждый элемент в списке состоит из

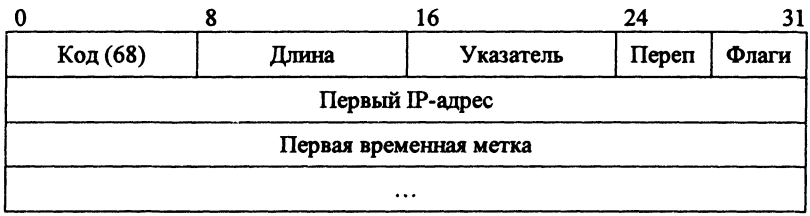


Рис. 5.25. Формат опции временных меток

двух 32-битных частей: IP-адреса маршрутизатора, заполнившего этот элемент, и 32-битового целого числа – временно́й метки. На рис. 5.25 приведен формат опции временны́х меток. На этом рисунке поля «Длина» и «Указатель» использованы для указания длины зарезервированного места и местонахождения следующего неиспользованного слота (как в опции записи маршрута). 4-битовое поле «Переп» содержит целое число шлюзов, которые не смогли записать временные метки из-за слишком маленького размера опции. Значение в 4-битовом поле «Флаги» определяет точный формат опции и говорит маршрутизаторам, как записывать временны́е метки. Допускаются следующие значения поля «Флаги»:

0 – только запись временных меток, IP-адреса опускаются;

1 – указывать перед каждой временной меткой IP-адрес (формат, показанный на рис. 5.25);

3 – IP-адреса указывает отправитель, маршрутизатор только записывает временную метку, если следующий IP-адрес в списке соответствует IP-адресу маршрутизатора.

Временны́е метки определяют время и дату, когда маршрутизатор обрабатывал дейтаграмму, и выражаются в миллисекундах после полуночи по Гринвичу. Если стандартное представление времени невозможно, маршрутизатор может использовать любое представление локального времени при условии, что он устанавливает старший бит в поле временной метки. Конечно, временные метки, записываемые независимыми компьютерами, не всегда согласованы, даже если представлены во времени по Гринвичу; каждая машина сообщает время согласно своим локальным часам, а часы могут идти по-разному. Поэтому, временные метки всегда рассматриваются как приблизительные оценки, независимо от их представления.

Может показаться странным, что опция временных меток включает механизм, заставляющий маршрутизатор записывать их IP-адреса вместе с временными метками, так как опция записи маршрута обеспечивает эту возможность. Тем не менее, запись IP-адресов вместе с временными метками позволяет избежать неоднозначности. Одновременная запись маршрута с временными метками также полезна потому, что она позволяет приемнику узнать точно, какой путь прошла дейтаграмма.

Обработка опций при фрагментации. При фрагментации дейтаграммы маршрутизатор повторяет некоторые IP-опции во всех фрагментах, в то время как другие помещаются только в один фрагмент. Например, рассмотрим оп-

цию, используемую для записи маршрута дейтаграммы. При передаче каждый фрагмент будет обрабатываться как независимая дейтаграмма, поэтому не гарантировано, что все фрагменты будут следовать по одному и тому же пути к месту назначения. Если все фрагменты содержат опцию записи маршрута, получатель может получить свой список шлюзов от каждого фрагмента. Он не сможет создать одного списка для собранной дейтаграммы. Поэтому, стандарт *IP* определяет, что опция записи маршрута должна копироваться только в один из фрагментов.

С другой стороны, рассмотрим, например, опцию маршрутизации источника, которая определяет, как должна передаваться дейтаграмма через Интернет. Информация о маршрутизации источника должна находиться в заголовках всех фрагментов, иначе фрагменты не будут следовать указанным путем. Поэтому, поле кода для маршрутизации источника указывает, что эта опция должна копироваться во все фрагменты.

Развитие межсетевого протокола IPv4

В июле 1992 г. Тематическая группа по технологии Интернет (IETF) выступила с инициативой на разработку требований к протоколам семейства TCP/IP нового поколения, названным IP Next Generation (IPng).

Одна из главных причин, почему IETF взялась за усовершенствование протокола IPv4, состояла в стремительном росте Интернета. Несмотря на то, что пространство адресов еще не исчерпано, потребность увеличения числа IP-адресов также диктуется тем фактом, что имеет место резервирование адресов блоками фиксированной величины, например, компания получает блок адресов класса B из 65000 уникальных адресов IPv4. Даже если компания использует не все зарезервированные адреса (что весьма вероятно) никто другой ими воспользоваться не может. Осознание этого факта послужило основным стимулом для разработки новой версии межсетевого протокола IPv6.

После обсуждения нескольких концепций в январе 1995 г. получила одобрение Группа управления технологией Интернет (IESG) и опубликовала запрос на комментарии и предложения (RFC 1752) «Рекомендации для протокола TP нового поколения». Данный документ охватывает базовые требования к IPng, описывает форматы заголовков пакетов, указывает подходы к организации адресного пространства и маршрутизации и содержит основные принципы построения средств обеспечения безопасности. Другие RFC-документы описывают технические подробности протокола, получившего название IPv6. К этим документам относятся: «Спецификации IPv6» (RFC 1883), «Архитектура адресного пространства IPv6» (RFC 1884), «Управление распределением адресного пространства IPv6» (RFC 1881), «Спецификация управляющего протокола для IPv6 (ICMPv6)» (RFC 1885), «Расширения DNS для поддержки IPv6»(RFC 1886) и др.

Рассмотрим основные дополнительные возможности протокола IPv6.

Расширенное адресное пространство. Одной из основных отличительных черт IPv6 является использование 128-разрядного адресного пространства по сравнению с 32-разрядным адресным пространством IPv4. Увеличение размера адреса с 32 до 128 бит позволяет не только существенно расширить адресное пространство, но и ввести больше иерархических уровней, чем адреса сети, подсети и рабочей станции в IPv4. Аналогично классической схеме адресации в IPv4, адрес IPv6 идентифицирует подключенный к сети интерфейс, а не компьютер. Основным отличием является тот факт, что интерфейс IPv6 не только может, но и должен иметь столько адресов, сколько это необходимо для обеспечения маршрутизации или сетевого управления. Адреса IPv6 принадлежат одной из следующих категорий: unicast, multicast и anycast. Unicast означает адрес в привычном смысле значения этого понятия. Данные адреса идентифицируют в точности один интерфейс в сфере своего действия и предназначены для информационного обмена точка-точка. Категория multicast идентифицирует адреса группы интерфейсов и предназначена для групповой рассылки информации. Пакет данных, посланный по такому адресу, должен быть доставлен по каждому из адресов интерфейсов, входящих в идентифицируемую группу. Адреса anycast также представляют группу интерфейсов, однако они доставляют информацию только на ближайший интерфейс из идентифицируемой группы.

Нотация адресов IPv6 представляет собой разделенные на 8 групп 16-битовые числа, записываемые в шестнадцатеричной системе счисления, например 0123:4567:89*AB*:*CDEF*:0123:4567:89*AB*:*CDEF*. При записи адреса в целях экономии места принято опускать незначащие нули.

Для уменьшения нагрузки на маршрутизаторы каждый IP-адрес должен не просто указывать место назначения, но и содержать достаточно информации для определения маршрута доставки пакетов. Один из способов достижения этой цели заключается в территориально-централизованном подходе к начальному распределению IP-адресов и установлению жесткой зависимости между всеми уровнями организаций-поставщиков услуг и их клиентами. Данный принцип был заложен на ранней стадии разработки спецификаций IPv6 и претерпел в настоящее время некоторые несущественные изменения. Необходимо заметить, что описываемая схема начального распределения адресов, называемая «aggregatable global unicast addresses», описывает лишь одну восьмую часть адресного пространства IPv6. Остальные адреса либо зарезервированы под определенные нужды, либо еще не распределены (доля последних составляет около 70 % всего адресного пространства). Формат адреса IPv6 представлен на рис. 5.26. Первый компонент адреса IPv6 является префиксом «aggregatable global unicast addresses» и имеет фиксированное значение (001). Второй компонент называется Агрегат данных высшего уровня (TLA – Top Level Aggregator). Согласно начальному плану распределения IP-адресов требовалось выделить фиксированные префиксы для трех основных регистров: Internet NIC (Network Information Center), обслуживающего Северную Америку,

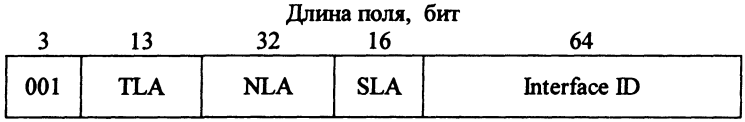


Рис. 5.26. Формат адреса *IPv6*

NCC (Network Coordination Center), координирующий деятельность ассоциации европейских сетей RIPE и APNIC, представляющий страны Азии и Тихого океана. Префиксы TLA присваивают ограниченному числу поставщиков услуг, которые, в свою очередь, сами назначают адреса своим клиентам.

Третий компонент адреса – Агрегат данных следующего уровня (NLA – Next Level Aggregator) – представляет собой гибкую структуру для использования сложившейся иерархии организаций – поставщиков услуг. Путем иерархического разбиения отведенного для NLA адресного пространства можно эффективно распределять сетевые адреса и управлять маршрутизацией потоков данных в пределах, контролируемых национальным или территориальным регистром.

Четвертый компонент адреса называется Агрегат данных уровня станции (SLA – Site Local Aggregator) и предназначен для назначения рабочей станции. При этом адрес рабочей станции выступает в роли атома системы адресов IPv6: при любом изменении полного адреса (например, в результате смены поставщика услуг Интернета) модификации подлежат только поля TLA и NLA. Компоненты SLA и Interface ID включают в себя MAC-адрес спецификации IEEE 802 и должны оставаться неизменными, что обеспечит глобально-уникальное именование активного сетевого оборудования.

Таким образом, расширение адресного пространства позволяет исключить необходимость преобразования сетевых адресов и предоставляет возможность использования различных типов адресов, например IPX. Автоматическая конфигурация адресов представляет собой одну из важнейших практических технологий в IPv6. Она не только избавляет от необходимости назначать новые адреса вручную, но и упрощает изменение ранее назначенных адресов.

IPv6 включает также поддержку мобильного IP для обеспечения маршрутизации между беспроводными и наземными сетями. Мобильное устройство сохраняет свой исходный адрес, но при этом оно получает второй адрес с информацией о местонахождении.

Усовершенствование маршрутизации. Для увеличения производительности маршрутизации в IPv6 применен новый формат заголовков пакетов. Новшество состоит в использовании меньшего, чем у IPv4, количества полей заголовка пакета, соответствующего сетевому уровню, и применении полей фиксированной длины. Большинство дополнительных полей вынесены в так называемые «опциональные заголовки», что делает возможным обработку маршрутизаторами меньшего количества обязательной информации. Кроме этого, IPv6 не предусматривает произведение дефрагментации пакетов маршрутизаторами. Эти функции должны выполняться только в точке отправления пакета.

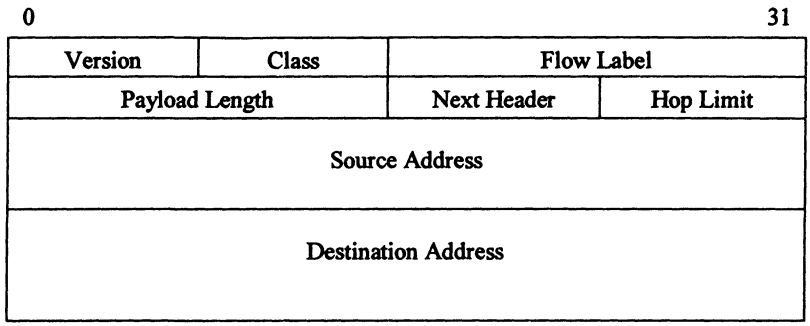


Рис. 5.27. Формат заголовка IPv6

Чтобы лучше понять, как информация в заголовке влияет на производительность маршрутизации, рассмотрим формат заголовка IPv6 (рис. 5.27). Заголовок пакета IPv6 состоит из 64-битового служебного поля и двух 128-битовых адреса источника и назначения, общим размером 40 байт (в IPv4 длина заголовка пакета сетевого уровня составляет 20 байт, не считая необязательного поля опций). В отличие от пакета IPv4, содержащего в заголовке 10 служебных полей и поле опций, пакет IPv6 состоит только из 6 полей.

Единственным общим полем протокола IPv6 и IPv4 является версия пакета, содержащая номер протокола. Идея разработчиков состоит в разделении на наиболее низком из возможных уровней потоков IP-пакетов различных версий. Так, в сетях Ethernet инкапсулированный фрейм IPv6 имеет тип 86DD, а IPv4 – 8000.

Отсутствие поля контрольной суммы заголовка пакета IPv6 напрямую обусловлено требованием сокращения накладных расходов на маршрутизацию. Действительно, поскольку на пути своего следования IP-пакет не должен претерпевать изменения, нет необходимости пересчитывать контрольную сумму заголовка. В то же время, отсутствие механизмов распознавания ошибок среды передачи может повлечь серьезные проблемы с определением путей доставки пакетов. Тем не менее, риск нераспознанной модификации заголовка пакета является незначительным, поскольку большинство процедур инкапсуляции IP-пакетов на уровне доступа к среде передачи используют контрольное суммирование. Это относится и к информационному обмену в локальных сетях (обязательные контрольные суммы описываются стандартом IEEE-803), и к сетям ATM (уровень AAL), и к передаче данных по коммутируемым каналам связи (процедура разбиения на фреймы протокола PPP).

Фиксированный размер заголовка пакета IPv6 и поле Payload Length (длина полезной нагрузки) полностью эквивалентны полям пакета IPv4 IHL (длина заголовка пакета) и Total Length (общая длина пакета). Поля Class и Flow Label управляют доставкой информации и замещают поле Type of Service пакета IPv4. Изменение поля «Время жизни» пакета IPv4 (TTL – Time-to-Live) на ограничение количества промежуточных узлов доставки пакета IPv6 (Hop Limit)

Next Header	Header Extension Length	Routing Type	Segments Left
Специфические для этого типа данные			

Рис. 5.28. Формат заголовка маршрутизации

фактически демонстрирует принципиально новый подход к управлению временем жизни пакета в сети Ethernet. Невозможность оценить реальное время нахождения пакета в канале связи и точное время обработки пакетов учтено введением нового поля в IPv6, где время пребывания в сети ограничивается числом промежуточных узлов.

Отправитель IPv6 использует заголовок маршрутизации для указания транзитных узлов, через которые пакет должен пройти на пути к адресату (рис. 5.28).

Поле Next Header (следующий заголовок) в заголовке маршрутизации сообщает о том, какой заголовок следует после заголовка маршрутизации. Поле Header Extension Length (длина расширения заголовка) – 8-значное целое число, выражающее длину заголовка маршрутизации в блоках из восьми октетов (исключая первые восемь октетов).

Поле Routing Type (тип маршрутизации) – 8-разрядный идентификатор специфической разновидности заголовка маршрутизации. Раздел Segments Left (оставшиеся сегменты) сообщает о числе оставшихся предопределенных транзитных узлов на пути пакета к адресату.

Формат поля специфических для указанного типа данных приведен в Routing Type. Длина поля данных такова, что общая длина заголовка маршрутизации составляет целое число, кратное восьми октетам. Заголовки маршрутизации такого типа прекрасно подходят для некоторых видов пакетов. Если длина пакета превышает предельное допустимое для сети значение (MTU – Maximum Transmission Unit), то пакет делят на фрагменты, каждый из которых передают как отдельный пакет.

При такой схеме отправители IPv6 используют заголовки фрагментов. Если в IPv4 фрагментация выполняется маршрутизаторами вдоль пути передачи пакета, то в IPv6 она выполняется только на отправителе.

Как известно, уменьшение нагрузки на маршрутизаторы можно достигнуть путем уменьшения объема анализируемой ими информации сетевого уровня. Данный подход нашел свое отражение в изменении формата заголовка IP-пакетов путем введения опциональных заголовков. Опциональные заголовки служат для указания специального режима обработки информации и могут не обрабатываться маршрутизаторами вообще или использоваться только в определенных случаях. Введение опциональных заголовков в IPv6 позволило полностью отказаться от использования поля «Опции» (Options) пакетов IPv4.

Спецификация IPv6 позволяет использовать произвольное число опциональных полей между заголовком сетевого уровня и полезными данными пакета. Содержимое опционального заголовка интерпретируется в соответствии с его типом. При этом каждый опциональный заголовок содержит тип следующего заголовка или полезных данных пакета.

Примерами опциональных заголовков являются аутентифицирующий заголовок и заголовок инкапсулированных зашифрованных данных.

Управление доставкой информации. Протокол IPv6 позволяет отмечать соответствие конкретного пакета определенным условиям его передачи, заданным отправителем. В результате достигается регулирование скорости передачи определенных потоков данных, что позволяет обеспечивать эффективную поддержку специальных протоколов (например, видео в режиме реального времени и др.). За счет назначения приоритетов передачи данных по определенным протоколам появилась возможность гарантировать первоочередность обработки наиболее критической информации и предоставления важным данным всей полосы пропускания канала связи. Другие особенности IPv6 позволяют протоколам этого семейства обеспечивать одновременную многоадресную доставку информации, что находит применение в рассылке информации «по подписке» или «по требованию», а также в других приложениях.

Средства обеспечения безопасности. Протокол IPv6 предоставляет возможности защиты от атак, связанных с подменой исходных адресов пакетов, и от несанкционированного доступа к полям данных пакетов. Это осуществляют за счет применения алгоритмов аутентификации и шифрования.

Протокол ICMP

Межсетевой протокол управляющих сообщений (ICMP – Internet Control Message Protocol) разработан для того, чтобы маршрутизаторы в Интернете сообщали об ошибках или предоставляли информацию о нестандартных условиях работы сети. Он является необходимой частью протокола IP. И обеспечивает обратную связь, оповещение отправителя данных о проблемах, возникающих в коммуникационном оборудовании.

Протокол ICMP – это механизм сообщения об ошибках. Он обеспечивает маршрутизаторам, обнаруживающим ошибки, способ сообщения об ошибке первоначальному источнику. Хотя спецификация протокола определяет допустимые способы использования ICMP и предлагает варианты возможных действий в ответ на ошибки, ICMP не специфицирует полностью действия, которые необходимо предпринять в ответ на все возможные ошибки. Таким образом, ICMP только сообщает о возникших ошибках первоначальному источнику; источник сам должен связать ошибки с конкретными прикладными программами и предпринять действия по исправлению ошибок.

Протокол ICMP выполняет следующие основные функции:

- обмен тестовыми сообщениями для выяснения наличия и активности узлов сети;
- анализ достижимости узлов и сброс пакетов, направленных к недостижимым узлам;
- изменение маршрутов (Redirect);
- уничтожение пакетов с истекшим временем жизни (Time-To-Live);
- синхронизация времени в узлах сети;
- управление трафиком (регулирование частоты отправки пакетов).

С точки зрения уровневых протоколов, ICMP является частью сетевого уровня. Но по отношению к IP ICMP протокол более высокого уровня, так как

он пользуется услугами IP для доставки своих сообщений. Другими словами, каждое сообщение ICMP передается по сети внутри IP-дейтаграммы.

ICMP-сообщения бывают двух видов: сообщение-запрос и сообщение об ошибке. Сообщение об ошибке тесно связано с породившей его дейтаграммой и всегда содержит заголовок этой IP-дейтаграммы и первые 64 бит ее данных. Это необходимо для того, чтобы узел-источник смог более точно проанализировать причину ошибки, так как все прикладные протоколы стека TCP/IP содержат наиболее важную информацию для анализа в первых 8 байт своего сообщения. Сообщения-запросы передают информацию об определенной сети и об определенном компьютере или их используют для диагностических целей.

IP-пакеты с сообщениями ICMP передаются по сети «на общих основаниях», без приоритетов, поэтому они тоже могут теряться. В загруженной сети они могут вызвать дополнительную загрузку маршрутизаторов, когда потеря сообщения об ошибке приводит к порождению нового сообщения и т. д., пока канал связи не исчерпает своей пропускной способности. Для того чтобы предотвратить подобные ситуации, в спецификации четко определены правила, руководствуясь которыми компьютер может решить, передавать его ICMP-сообщение или нет.

Правило 1: потеря пакета с ICMP-сообщением никогда не генерирует нового ICMP-сообщения.

Правило 2: сообщения об ошибке никогда не генерируются в ответ на IP-дейтаграммы с широковещательными или групповыми адресами, чтобы не вызвать полную перегрузку сети – широковещательный шторм (*broadcast storm*).

Правило 3: при повреждении фрагментированной дейтаграммы, ICMP-сообщение отправляют только после первого поврежденного фрагмента (так как источник все равно повторит передачу всей дейтаграммы целиком).

Доставка ICMP-пакетов требует двух уровней инкапсуляции. ICMP-пакеты инкапсулируются внутрь IP-дейтаграммы, которая сама передается по каждой физической сети в поле данных кадра (рис. 5.29).

Несмотря на то, что сообщения ICMP инкапсулируются и посылаются, используя IP, ICMP не считают протоколом более высокого уровня – он является

Рис. 5.29. Два уровня инкапсуляции сообщения ICMP

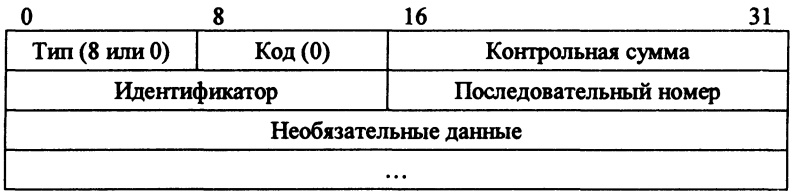


Рис. 5.30. Формат пакета ICMP

необходимой частью IP. Протокол IP необходим для транспортировки сообщений ICMP, так как им, чтобы достичь своего конечного назначения, надо пересечь несколько физических сетей. Поэтому, они не могут быть доставлены только с помощью физической передачи.

Формат ICMP-пакетов. Хотя каждое сообщение ICMP имеет свой собственный формат, все они начинаются с трех одинаковых полей: 8-битового целочисленного поля «Тип», идентифицирующего сообщение, 8-битового поля «Код», обеспечивающего более точную информацию о типе сообщения, и 16-битового поля «Контрольная сумма» (рис. 5.30). Помимо того, сообщения ICMP, сообщающие об ошибках, всегда включают заголовок и первые 64 бит данных дейтаграммы, вызвавшей ошибку. Это необходимо для того, чтобы узел-отправитель смог более точно проанализировать причину ошибки, так как все протоколы прикладного уровня стека TCP/IP содержат наиболее важную информацию для анализа в первых 64 бит своих сообщений.

Сетевые программы распознают ICMP-сообщения по двум признакам: значению поля «Тип» и значению поля «Код». Контрольная сумма вычисляется не только для ICMP-заголовка, но и для всего сообщения.

Таблица 5.12. **Типы сообщений ICMP**

Тип сообщения ICMP	Описание
0	Ответ на эхо (Echo Reply)
3	Узел назначения недостижим (Destination Unreachable)
4	Подавление источника (Source Quench)
5	Перенаправление маршрута (Redirect)
8	Запрос эха (Echo Request)
9	Информация о маршрутизаторах (Router Advertisement)
10	Регистрация маршрутизатора (Router Solicitation)
11	Лимит времени для дейтаграммы превышен (Time Exceeded for a Datagramm)
12	Проблема с параметром пакета (Parameter Problem on a Datagramm)
13	Запрос метки времени (Timestamp Request)
14	Ответ для метки времени (Timestamp Reply)
15	Запрос информации (не действует)
16	Ответ на запрос информации (не действует)
17	Запрос маски адреса (Address Mask Request)
18	Ответ на запрос маски адреса (Address Mask Reply)

Типы сообщений ICMP представлены в табл. 5.12. Рассмотрим каждое из этих сообщений и его формат подробнее.

Тестирование достижимости места назначения. Протоколы TCP/IP обеспечивают средства, помогающие сетевым администраторам или пользователям идентифицировать проблемы в сети. Пользователь в качестве одного из широко используемых средств отладки применяют команду, которая вызывает сообщения ICMP запроса эха и ответа эха. Компьютер или маршрутизатор посылает сообщение запроса эха указанному месту назначения. Любая машина, получившая запрос эха, генерирует ответ на эхо и возвращает его первоначальному отправителю. Этот запрос содержит необязательную область данных; ответ содержит копию данных, посланных в запросе. Запрос эха и связанный с ним ответ можно использовать для проверки достижимости назначения и его способности отвечать на запросы. Так как и запрос эха, и ответ на него передаются в IP-дейтаграммах, успешный прием ответа свидетельствует о работоспособности основных частей транспортной системы. Во-первых, программное обеспечение IP на машине источника выполнило маршрутизацию дейтаграммы. Во-вторых, промежуточные маршрутизаторы между источником и получателем работоспособны и корректно маршрутизируют дейтаграммы. В-третьих, машина получателя работает (по крайней мере, она обрабатывает прерывания) и программное обеспечение, как IP, так и ICMP, выполняет свои функции. И, наконец, таблицы маршрутов в маршрутизаторах на всем обратном пути корректны.

Во многих системах команда, которую пользователи вызывают для посылки запроса эха ICMP, называется *ping*. Усложненные версии этой программы посылают серии запросов эха ICMP, принимают ответы и выдают статистику о потерях дейтаграмм. Они позволяют пользователю указывать длину посылаемых данных и интервалы времени между запросами. Менее сложные версии просто посылают запрос эха ICMP и ждут ответа.

Формат сообщения запроса эха и ответа эха. Средства для тестирования достижимости узлов сети представляют собой очень простой эхо-протокол, включающий обмен двумя типами сообщений: эхо-запрос и эхо-ответ. Компьютер или маршрутизатор посылают по интерсети эхо-запрос, в котором указывают IP-адрес узла, достижимость которого нужно проверить. Узел, получающий эхо-запрос, формирует и отправляет эхо-ответ и возвращает сообщение узлу – отправителю запроса. В запросе могут содержаться некоторые данные, которые должны быть возвращены в ответе.

Рис. 5.30 иллюстрирует формат сообщений запроса эха и ответа на запрос эха. Поле «Необязательные данные» имеет переменную длину и содержит данные, которые надо вернуть отправителю. Ответ на эхо всегда возвращает те же самые данные, что были получены им в запросе. Поля «Идентификатор» и «Последовательный номер» отправитель использует для проверки соответствия ответов запросам. Значение поля «Тип» определяет, является ли сообщение запросом (8) или ответом (0).

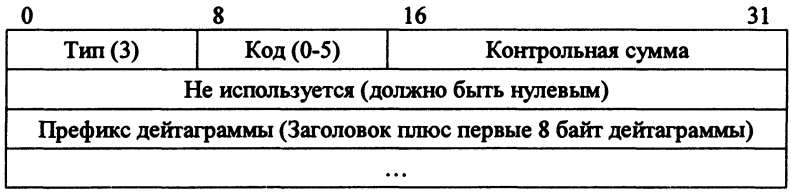


Рис. 5.31. Формат сообщения о недостижимости назначения

Сообщения о недостижимости назначения. Когда маршрутизатор не может доставить IP-дейтаграмму, он посылает сообщение «назначение недостижимо» первоначальному отправителю, используя формат, приведенный на рис. 5.31. Поле «Код» в сообщении о недостижимости назначения содержит целое число, которое описывает причину. Возможные значения представлены в табл. 5.13.

Таблица 5.13. **Коды сообщений о недостижимости**

Код сообщения	Пояснения
0	Сеть недостижима
1	Компьютер недостижим
2	Протокол недостижим
3	Порт недостижим
4	Необходима фрагментация
5	Ошибка при маршрутизации источника
6	Сеть назначения неизвестна
7	Компьютер назначения неизвестен
8	Компьютер источника изолирован
9	Взаимодействие с сетью назначения административно запрещено
10	То же с компьютером назначения
11	Сеть недостижима из-за класса обслуживания
12	Компьютер недостижим из-за класса обслуживания

Хотя протокол IP является механизмом ненадежной доставки, дейтаграммы не уничтожаются просто так. Всякий раз, когда ошибка мешает маршрутизатору произвести маршрутизацию или доставку дейтаграммы, маршрутизатор посылает сообщение о недостижимости назначения его источнику, а затем уничтожает дейтаграмму. Ошибки недостижимости сети обычно являются следствием ошибок маршрутизации; ошибки недостижимости компьютера – следствие ошибок при доставке.

Назначения могут быть недостижимыми из-за того, что оборудование было временно неработоспособно, отправитель указал несуществующий адрес назначения или (в редких случаях) у маршрутизатора не указано маршрута к сети назначения. Необходимо отметить, что не все подобные ошибки можно обнаружить.

Если дейтаграмма содержит опцию маршрутизации источника с некорректным маршрутом, то это может привести к появлению сообщения об ошибке маршрутизации источника. Если шлюзу нужно фрагментировать дейтаграмму, но установлен бит «не фрагментировать», то шлюз посылает сообщение «требуется фрагментация» обратно источнику.

Управление потоком дейтаграмм и переполнение сети. Так как IP-протокол не устанавливает соединения, то маршрутизаторы не могут резервировать память или коммуникационные ресурсы до получения дейтаграмм. В результате, трафик может вызвать перегрузку маршрутизаторов, ситуацию, называемую переполнением сети (congestion). Переполнение сети происходит по двум совершенно разным причинам. Во-первых, высокоскоростной компьютер может генерировать трафик быстрее, чем сеть может передавать его. Например, представим суперкомпьютер, генерирующий межсетевой трафик. Дейтаграммам, посылаемым им, может потребоваться передача, в конечном счете, по медленной глобальной сети (WAN), хотя сам суперкомпьютер может быть присоединен к высокоскоростной LAN. Переполнение будет возникать в маршрутизаторе, присоединенном к глобальной сети, так как дейтаграммы будут прибывать быстрее, чем их можно послать. Во-вторых, если большому числу компьютеров одновременно нужно посылать дейтаграммы через один маршрутизатор, этот маршрутизатор может оказаться переполненным, хотя ни один источник в отдельности не вызывает эту проблему.

Когда дейтаграммы прибывают на шлюз или маршрутизатор быстрее, чем он успевает их обрабатывать, он временно ставит их в очередь в своей памяти. Если эти дейтаграммы создают небольшую пиковую нагрузку при передаче дейтаграмм, то такая буферизация решает проблему. Если же трафик продолжает поступать, то, в конечном счете, маршрутизатор или шлюз займет всю память под очередь и вынужден будет удалять новые прибывающие дейтаграммы. Тогда машина для выхода из состояния переполнения использует сообщения о подавлении источника.

Сообщение о подавлении источника требует от источника уменьшить скорость передачи дейтаграмм. Обычно переполненные маршрутизаторы посылают по одному сообщению о подавлении источника на каждую удаляемую дейтаграмму или используют более сложные технологии выхода из переполнения. Формат подавления источника представлен на рис. 5.32. Помимо обычных полей ICMP «Тип», «Код» и «Контрольная сумма» и неиспользуемого 32-битового поля, сообщения о подавлении источника имеют поле, содержащее

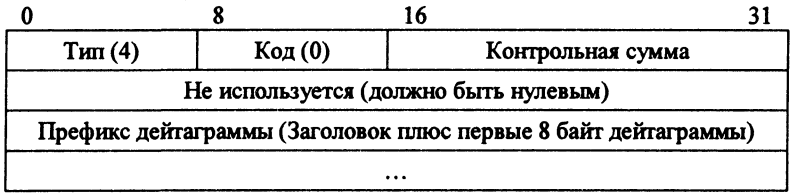


Рис. 5.32. Формат сообщения о подавлении источника ICMP

префикс дейтаграммы. Как и в других сообщениях об ошибках ICMP поле префикса дейтаграммы содержит префикс дейтаграммы, вызвавшей этот запрос подавления источника.

Сообщения ICMP, вызывающего эффект, обратный подавлению источника, не существует. Вместо этого, компьютер, принявший сообщения о подавлении источника от некоторой машины, снижает скорость, с которой он посылает ей дейтаграммы. Это происходит до тех пор, пока к нему не перестанут приходить сообщения о подавлении источника. Затем он постепенно увеличивает скорость пока снова не получит сообщения о подавлении источника.

Перенаправление маршрута. Маршрутные таблицы у компьютеров обычно статические, так как их конфигурирует администратор сети, а у маршрутизаторов – динамические, формируемые автоматически с помощью протоколов обмена маршрутной информации. Поэтому с течением времени при изменении топологии сети маршрутные таблицы компьютеров могут устаревать.

При изменении топологии сети таблицы маршрутизации в маршрутизаторе или компьютере могут стать некорректными. Изменение может быть временным (например, нужно заменить неисправное оборудование) или постоянным (например, когда в межсетевое взаимодействие включается новая сеть). Маршрутизаторы периодически обмениваются информацией о маршрутизации, чтобы отслеживать изменения в сети и своевременно менять маршруты. Для корректировки поведения компьютеров маршрутизатор может использовать сообщение протокола ICMP, называемое «перенаправлением» (Redirect), запрашивающее изменение маршрута в таблице маршрутизации компьютера.

Механизм перенаправления протокола ICMP позволяет компьютерам содержать в конфигурационном файле только IP-адреса его локальных маршрутизаторов. С помощью сообщений о перенаправлении маршрутизаторы будут сообщать компьютеру всю необходимую ему информацию о том, какому маршрутизатору следует отправлять пакеты для той или иной сети назначения, т. е. маршрутизаторы передадут компьютеру нужную ему часть их таблиц маршрутизации.

Преимуществом схемы перенаправления ICMP является ее простота: она позволяет компьютеру знать вначале адрес только одного маршрутизатора в локальной сети. Этот начальный маршрутизатор возвращает сообщение ICMP о перенаправлении всякий раз, когда компьютер посылает дейтаграмму, для которой существует лучший маршрут. Таблица маршрутизации компьютера останется маленькой, но содержит оптимальные маршруты для всех используемых назначений.

Сообщения о перенаправлении, тем не менее, не решают проблему распространения информации о маршрутах полностью, так как они предназначены только для взаимодействия между маршрутизатором и компьютером в одной физической сети. Каждое сообщение о перенаправлении содержит 32-битовое поле «IP-адрес маршрутизатора» и поле «Префикс дейтаграммы», как это показано на рис. 5.33.

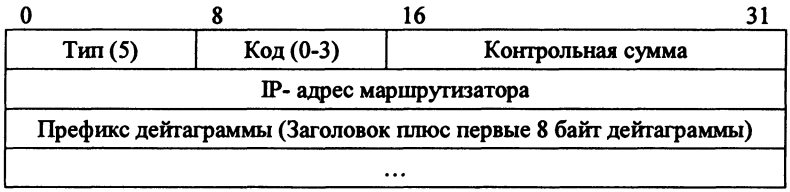


Рис. 5.33. Формат сообщения о перенаправлении ICMP

Поле «Межсетевой адрес маршрутизатора» содержит IP-адрес маршрутизатора, который должен использовать компьютер при отправлении дейтаграммы к назначению, указанному в заголовке дейтаграммы. Поле «Префикс дейтаграммы» содержит заголовок IP и следующие 8 байт дейтаграммы, которая привела к появлению этого сообщения. Поэтому компьютер, принимающий сообщение о перенаправлении ICMP, должен выделить адрес назначения дейтаграммы из префикса дейтаграммы. Поле «Код» в сообщении о перенаправлении ICMP более конкретно указывает, как интерпретировать адрес назначения, при этом значения имеют следующий смысл: 0 – перенаправление дейтаграмм для этой сети (устарело), 1 – перенаправление дейтаграмм для этого компьютера, 2 – перенаправление дейтаграмм для этого типа сервиса и сети, 3 – перенаправление дейтаграмм для этого типа сервиса и компьютера. Напомним, что каждый заголовок IP указывает тип сервиса, используемого при маршрутизации. Как правило, маршрутизаторы посылают запросы переназначения ICMP только на компьютеры, а не на другие маршрутизаторы.

Изменение маршрута является одной из наиболее интересных функций протокола ICMP – по существу, это один из механизмов автоматической оптимизации доставки пакетов и адаптации сетей TCP/IP к изменениям топологии.

Запросы «Информация о маршрутизаторах» (типы 9 и 10).

Информация о маршрутизации находится в местных конфигурационных файлах и загружается оттуда при запуске компьютера. Чтобы таблица маршрутизации не содержала устаревших данных она обновляется динамически. ICMP-протокол реализует один из способов ее обновления.

Существует 2 типа сообщений маршрутизаторов:

- 9 – информация о маршрутизации;
- 10 – регистрация маршрутизатора.

Всякий раз, когда компьютер запускают, он генерирует сообщения о регистрации. В ответ маршрутизаторы, находящиеся в той же локальной сети, посылают сообщения с информацией о маршрутизации, позволяющие правильно сконфигурировать маршрутную таблицу.

Формат сообщения «Информация о маршрутизации» (тип 9) описан в RFC1256 (рис. 5.34).

0	8	16	31
Тип (9)	Код (0)	Контрольная сумма	
Количество адресов	Длина поля адреса	Время существования	
IP-адрес маршрутизатора 1			
Приоритет 1			
IP- адрес маршрутизатора 2			
Приоритет 2			
...			

Рис. 5.34. Формат сообщения «Информация о маршрутизации»

В одном ICMP-сообщении может содержаться описание нескольких адресов, количество которых указано в поле «Количество адресов». Поле «Размер адреса» задает длину адреса в 32-битовых словах. В настоящее время «Длина поля адреса» всегда равна 2.

Поле «Время существования» задает интервал времени, в течение которого информация еще не устарела. Как правило, это 1800 с.

Поле «Приоритет» указывает, какой из адресов следует использовать первым и более интенсивно. Как правило, чем больше значение поля, тем выше приоритет. Маршрутизаторы передают информационные сообщения широковещательно через случайные интервалы времени. Обычно через 450...600 с. Поле «Время существования» можно использовать для уведомления, что данный маршрутизатор выключается. При этом содержимое данного поля устанавливается равным 0.

Формат сообщения «Регистрация» (тип 10) представлен на рис. 5.35.

Запрос «Регистрация» передается 3 раза с интервалом 3 с при запуске маршрутизатора и продолжает (при необходимости) передаваться, пока маршрутизатор не получит информационного сообщения с нужной маршрутной информацией.

Обнаружение циклических или слишком длинных путей. Как было отмечено выше для защиты Интернета от перегрузок каждая дейтаграмма имеет счетчик времени жизни TTL (Time-To-Live). Маршрутизатор декрементирует счетчик времени жизни всякий раз, когда он обрабатывает дейтаграмму, и удаляет ее, когда счетчик становится нулевым.

0	8	16	31
Тип (10)	Код (0)	Контрольная сумма	
Указатель	Заполняется нулями		

Рис. 5.35. Формат сообщения «Регистрация»

0	8	16	31
Тип (10)	Код (0)	Контрольная сумма	
Указатель	Заполняется нулями		

Рис. 5.36. Формат сообщения «Лимит времени для дейтаграммы превышен»

Независимо от того, удалил ли маршрутизатор дейтаграмму из-за обнуления счетчика времени жизни или из-за превышения времени ожидания фрагментов дейтаграммы, он посылает сообщение ICMP «Лимит времени для дейтаграммы превышен» источнику дейтаграммы определенного формата (рис. 5.36).

Поле «Код» объясняет причину сообщения: 0 – превышено значение счетчика времени жизни; 1 – превышено время ожидания фрагмента при сборке.

Сообщения о других ситуациях. Когда маршрутизатор или компьютер сталкивается с проблемой, не укладывающейся в рамки описанных сообщений об ошибках ICMP (например, некорректный заголовок дейтаграммы), связанной с дейтаграммой, он посылает сообщение «Проблема с параметром пакета» первоначальному отправителю. Такую ситуацию может вызвать некорректность аргументов опции. Сообщение, формат которого показан на рис. 5.37, посылается только в том случае, если дейтаграмма должна быть удалена из-за этой ошибки. Для уточнения места ошибки в дейтаграмме отправитель использует поле «Указатель» в заголовке сообщения для идентификации октета в дейтаграмме, содержащего ошибку.

Синхронизация часов и оценка времени передачи. Стек протоколов TCP/IP включает несколько протоколов, которые могут использоваться для синхронизации часов. В сети для этого используется несколько технологий. Одна из простейших технологий реализуется сообщениями ICMP для получения значения времени от другой машины. Запрашивающая машина посылает сообщение ICMP «Запрос метки времени» другой машине, ожидая, что вторая машина вернет ей текущее значение времени. Принимающая машина возвращает «Ответ для метки времени» машине, выдавшей запрос. Рис. 5.38 иллюстрирует формат сообщений запроса и ответа временной метки. Поле «Тип» идентифицирует сообщение как запрос (13) или ответ (14); поля «Идентификатор» и «Последовательный номер» используют источник для связи между ответами и запросами. Оставшиеся поля специфицируют времена, указанные в миллисекундах после полуночи, по Гринвичу. Поле «Временная метка отправителя» заполняет

0	8	16	31
Тип (12)	Код (0-1)	Контрольная сумма	
Указатель	Не используется (должно быть нулевым)		
Префикс дейтаграммы (Заголовок плюс первые 8 байт дейтаграммы)			
...			

Рис. 5.37. Формат сообщения «Проблема с параметром пакета»

0	8	16	31
Тип (13 или 14)	Код (0)	Контрольная сумма	
Идентификатор		Последовательный номер	
Префикс дейтаграммы (Заголовок плюс первые 8 байт дейтаграммы)			
Временная метка отправителя			
Временная метка приема			
Временная метка передачи			

Рис. 5.38. Формат сообщений «Запрос метки времени» и «Ответ для метки времени»

первоначальный отправитель перед передачей пакета, поле «Временная метка приема» заполняется сразу после приема запроса, а поле «Временная метка передачи» – непосредственно перед отправкой ответа.

Компьютеры используют эти три поля временных меток для определения ожидаемого времени передачи между ними и синхронизации своих часов. Так как ответ включает поле «Временная метка отправителя», компьютер может вычислить общее время, требуемое для передачи запроса к назначению, формирования ответа на него и возвращения ответа. Так как ответ содержит как время прихода запроса на удаленную машину, так и время выдачи ответа, компьютер может вычислить время передачи по сети, а на его основе – разницу между своими и удаленными часами. На практике бывает трудно точно оценить время передачи по сети, так как IP является технологией с негарантированной доставкой, дейтаграммы могут быть потеряны, задержаны или доставлены не по порядку, что ограничивает полезность сообщений ICMP о временных метках.

Сообщения запроса и ответа информации. Сообщения ICMP запроса информации и ответа информации (тип 15 и 16) в настоящее время устарели и их использовать не рекомендуется. Они предназначались для обнаружения компьютерами своих IP-адресов при загрузке. Сейчас для определения адреса используют протоколы RARP и BOOTP.

Получение маски подсети. Для применения адресации подсетей компьютеру надо знать, какие биты их 32-битного IP-адреса соответствуют физической сети, а какие – адресу компьютера. Информация, требуемая для интерпретации адреса, представляет собой 32-битовую величину, называемую маской подсети. Чтобы узнать маску подсети, используемую в локальной сети, машина может послать сообщение запроса маски адреса маршрутизатору и получить ответ маски адреса. Машина, формирующая запрос, может либо послать сообщение напрямую, если она знает адрес маршрутизатора, либо послать широковещательное сообщение, если не знает его. Рис. 5.39 иллюстрирует формат сообщений маски адреса. Поле «Тип» в сообщении маски адреса указывает, является ли сообщение запросом (17) или ответом (18). Ответ содержит маску адреса подсети в поле «Маска адреса». Как правило, поля «Идентификатор» и «Последовательный номер» позволяют машине связать ответ с запросом.

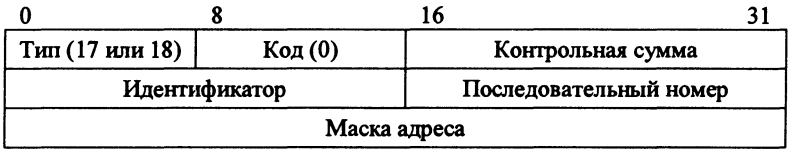


Рис. 5.39. Формат сообщений «Запрос маски адреса» и «Ответ на запрос маски адреса»

Недостатком протокола ICMP является сохраняющаяся возможность несанкционированной посылки ложного ICMP Redirect сообщения о смене маршрута от имени маршрутизатора.

Протоколы маршрутизации

Алгоритмы маршрутизации играют важную роль в IP-сетях. Главным параметром при маршрутизации пакета является IP-адрес его места назначения. Проблема оптимальной маршрутизации в современной сети Internet, насчитывающей уже давно более 10 млн узлов, весьма сложна. Протокол IP делит все машины на маршрутизаторы (Router) и обычные компьютеры (Host), последние, как правило, не рассылают свои маршрутные таблицы. Предполагается, что маршрутизатор владеет исчерпывающей информацией о правильных маршрутах, обычный же компьютер имеет минимальную маршрутную информацию (например, адрес маршрутизатора локальной сети) и все необходимые для решения этой проблемы данные получает из маршрутизатора.

Автономная система (AS) может содержать множество маршрутизаторов, но взаимодействие с другими AS она осуществляет только через один маршрутизатор, называемый пограничным (Border Gateway, именно они дали название протоколам BGP). Пограничный маршрутизатор необходим, когда автономная система имеет более одного внешнего трафика, в противном случае его функции выполняет порт внешнего подключения (Gateway). Если адресат достижим более чем одним путем, маршрутизатор должен сделать выбор маршрута на основании оценки маршрутов кандидатов. Обычно каждому сегменту, составляющему маршрут, присваивается некоторая оценка, например, транзитный маршрутизатор. Каждый протокол маршрутизации использует свою систему оценки маршрутов. Оценка сегмента маршрута называется *метрикой*.

Маршрутизатор может использовать два протокола маршрутизации одновременно: один – для внешних связей, другой – для внутренних. Все протоколы обмена маршрутной информацией стека TCP/IP относятся к классу адаптивных протоколов, которые, в свою очередь, разделены на две группы, каждая из которых связана с одним из следующих типов алгоритмов:

- дистанционно-векторный алгоритм (DVA – Distance Vector Algorithms),
- алгоритм состояния связей (LSA – Link State Algorithms).

В алгоритмах *дистанционно-векторного типа* каждый маршрутизатор периодически и широковещательно рассылает по сети вектор расстояний от себя до всех известных ему сетей. Под расстоянием обычно понимают число промежуточных маршрутизаторов через которые пакет должен пройти прежде, чем попадет в соответствующую сеть. Может использоваться и другая метрика, учитывающая не только число транзитных пунктов, но и время прохождения пакетов по связи между соседними маршрутизаторами. Получив вектор от соседнего маршрутизатора, каждый маршрутизатор добавляет к нему информацию об известных ему других сетях, о которых он узнал непосредственно (если они подключены к его портам) или из аналогичных объявлений других маршрутизаторов, а затем снова рассылает новое значение вектора по сети. В результате каждый маршрутизатор узнает информацию об имеющихся в интерсети сетях и о расстоянии до них через соседние маршрутизаторы.

Дистанционно-векторные алгоритмы хорошо работают только в небольших сетях. В больших сетях они засоряют линии связи интенсивным широковещательным трафиком, к тому же изменения конфигурации отрабатываются по этому алгоритму не всегда корректно, так как маршрутизаторы не имеют точного представления о топологии связей в сети, а располагают только обобщенной информацией – вектором дистанций, к тому же полученной через посредников. Работа маршрутизатора в соответствии с дистанционно-векторным протоколом напоминает работу моста, так как точной топологической картины сети такой маршрутизатор не имеет.

Наиболее распространенным протоколом, основанным на дистанционно-векторном алгоритме, является протокол RIP (Routing Information Protocol, RFC-1058, -1721-27), разработанный фирмой Xerox.

Алгоритмы состояния связей обеспечивают каждый маршрутизатор информацией, достаточной для построения точного графа связей сети. Все маршрутизаторы работают на основании одинаковых графов, что делает процесс маршрутизации более устойчивым к изменениям конфигурации. Широковещательная рассылка используется здесь только при изменениях состояния связей, что происходит в надежных сетях не так часто. Для того чтобы понять, в каком состоянии находятся линии связи, подключенные к его портам, маршрутизатор периодически обменивается короткими пакетами со своими ближайшими соседями. Этот трафик также широковещательный, но он циркулирует только между соседями и поэтому не так засоряет сеть.

Протокол маршрутизации RIP

RIP – внутренний протокол маршрутизации IGP (Interior Gateway Protocol, RFC-1074, -1371) определяет маршруты внутри автономной системы. Этот протокол маршрутизации предназначен для сравнительно небольших и относительно однородных сетей. В протоколе RIP сообщения инкапсулируются в UDP-дейтаграммы, при этом работает порт 520. В качестве метрики маршрутизации RIP использует число шагов (хопов) до цели. Если между отправителем и

приемником расположено три маршрутизатора, считается, что между ними четыре шага. Такой вид метрики не учитывает различий в пропускной способности или загруженности отдельных сегментов сети. Таблица маршрутизации RIP содержит по одной записи на каждую обслуживаемую машину. Запись обычно содержит следующие поля:

- сеть (IP-адрес сети);
- расстояние до этой сети;
- IP-адрес следующего маршрутизатора по пути к месту назначения;
- таймеры маршрута.

Вектором расстояний будем называть набор пар («Сеть», «Расстояние до этой сети»), извлеченный из маршрутной таблицы, а каждую пару этого набора – *элементом вектора расстояний.*

Существует две версии протокола RIP. RIP-1 – описан в документе RFC-1058. RIP-2 (RFC-1721-24, 1993 г.) – новая версия RIP, которая в дополнение к широковещательному режиму поддерживает групповую рассылку (multicast); позволяет работать с масками подсетей.

Формат сообщения протокола RIP показан на рис. 5.40.

Поле «Команда» (Command) определяет тип сообщения : 1 – запрос (request) на получение частичной или полной маршрутной информации; 2 – ответ (response), содержащий информацию о расстояниях из маршрутной таблицы отправителя; 3 – включение режима трассировки (устарело); 4 – выключение режима трассировки (устарело); 5,6 – зарезервированы для внутренних целей.

Поле «Версия» (Version) для RIP-1 равно 1 (для RIP-2 – 2).

Поле «Набор протоколов сети» (Address Family Identifier) определяет набор протоколов, которые используются в соответствующей сети (для Internet это поле имеет значение 2).

Поле «Расстояние до сети» (Metric) содержит целое число шагов (от 1 до 15) до данной сети. Если расстояние равно 16, то считается , что сеть недостижима.

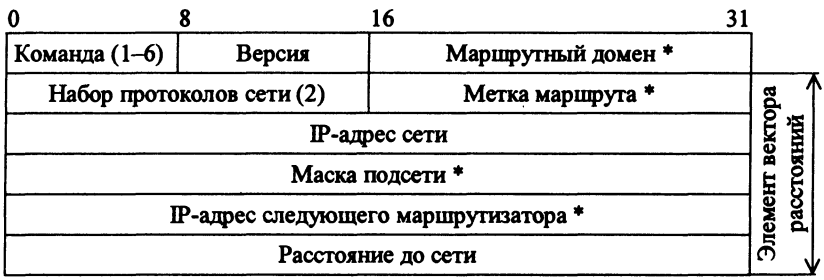


Рис. 5.40. Формат сообщения протокола RIP:

*** – поля, относящиеся ко второй версии протокола RIP**

(в сообщениях протокола первой версии эти поля должны быть обнулены)

Поле «Маршрутный домен» (Routing Domain) служит идентификатором RIP-системы, к которой принадлежит данное сообщение; часто это номер автономной системы, который используется, когда к одному физическому каналу подключены маршрутизаторы из нескольких автономных систем, в каждой автономной системе поддерживается своя таблица маршрутов. Поскольку RIP-сообщения рассылаются всем маршрутизаторам, подключенным к сети, требуется различать сообщения, относящиеся к «своей» и «чужой» автономным системам.

Поле «Метка маршрута» (Route Tag) выполняет роль метки для внешних маршрутов при работе с протоколами внешней маршрутизации.

Поле «Маска подсети» (Subnet Mask) – маска сети, адрес которой содержится в поле IP-адрес. RIP-1 работает только с классовой моделью адресов.

Поле «IP-адрес следующего маршрутизатора» (Next Hop) содержит адрес следующего маршрутизатора для данного маршрута, если он отличается от адреса маршрутизатора, пославшего данное сообщение. Это поле используют, когда к одному физическому каналу подключены маршрутизаторы из нескольких автономных систем и, следовательно, некоторые маршрутизаторы «чужой» автономной системы могут быть достигнуты напрямую, минуя пограничный маршрутизатор. Об этом пограничный маршрутизатор и объявляет в поле «IP-адрес следующего маршрутизатора». Адрес 0.0.0.0 в сообщении типа «ответ» обозначает маршрут, ведущий за пределы RIP-системы. В сообщении типа «запрос» этот адрес означает запрос информации о всех маршрутах (полного вектора расстояний). Указание в сообщении типа «запрос» адреса конкретной сети означает запрос элемента вектора расстояний только для этой сети – такой режим используют обычно только в отладочных целях. Аутентификация может производиться протоколом RIP-2 для обработки только тех сообщений, которые содержат правильный аутентификационный код. При работе в таком режиме первый 20-октетный элемент вектора расстояний, следующий непосредственно за первым 32-битным словом RIP-сообщения, является сегментом аутентификации. Его определяют по значению поля «Набор протоколов сети» (Address Family Identifier), равному в этом случае $(FFFF)_h$. Следующие 2 октета этого элемента определяют тип аутентификации, а остальные 16 октетов содержат аутентификационный код. Таким образом, в RIP-сообщении с аутентификацией может передаваться не 25, а только 24 элемента вектора расстояний, которые следуют за сегментом аутентификации. К настоящему моменту надежного алгоритма аутентификации для протокола RIP не разработано; стандартом определена только аутентификация с помощью обычного пароля (значение поля «Тип» равно 2).

Сообщение RIP состоит из 32-битного слова, определяющего тип сообщения и версию протокола (плюс «Маршрутный домен» в RIP-2), за которым следует набор из одного или более элементов вектора расстояний. Каждый элемент вектора расстояний занимает 5 слов (20 октетов) (см. рис. 5.40).

Максимальное число элементов вектора равно 25, если вектор длиннее, он может разбиваться на несколько сообщений. Таким образом, одно RIP-сообщение может содержать информацию о 25 маршрутах.

С точки зрения маршрутизации работа RIP-2 принципиально не отличается от первой версии протокола. Рассмотрим работу RIP-1 подробнее.

Алгоритм построения таблицы маршрутов. Для более наглядного представления алгоритма введем следующие обозначения.

- Строку в таблице маршрутов будем записывать в виде $A = 2 \rightarrow R3$. Это означает, что расстояние от данного маршрутизатора до сети A равно 2, а дейтаграммы, следующие в сеть A, следует пересылать маршрутизатору $R3$.
- Вектор расстояний будем записывать в виде ($A = 2$, $B = 1$). Это означает, что расстояние от данного маршрутизатора до сети A равно 2, до сети B равно 1.
- Расстояние до сети, к которой маршрутизатор подключен непосредственно, примем равным 1.

Каждый маршрутизатор, на котором запущен модуль RIP, периодически широковещательно распространяет свой вектор расстояний. Вектор распространяется через все интерфейсы (порты) маршрутизатора, подключенные к сетям, входящим в RIP-систему. Каждый маршрутизатор также периодически получает векторы расстояний от других маршрутизаторов. Расстояния в этих векторах инкрементируются (увеличиваются на 1), после чего сравниваются с данными в таблице маршрутов, и, если расстояние до какой-то из сетей в полученном векторе оказывается меньше расстояния, указанного в таблице, значение из таблицы замещается новым (меньшим) значением, а адрес маршрутизатора, приславшего вектор с этим значением, записывается в поле «Следующий маршрутизатор» в этой строке таблицы. После этого вектор расстояний данного маршрутизатора соответственно изменится.

Рассмотрим построения маршрутной таблицы на примере сети, представленной на рис. 5.41 (компьютеры в сетях не показаны). Рассмотрим процесс формирования таблицы маршрутов применительно к узлу $R1$.

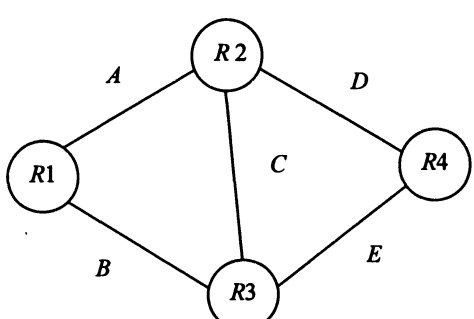


Рис. 5.41. Пример структуры RIP-системы:
$R1 - R4$ – маршрутизаторы; $A - E$ – сети

В начальный момент времени (например, после подачи питания на маршрутизаторы) таблица маршрутов в узле $R1$ (узел $R1$ знает только о тех сетях, к которым подключен непосредственно) выглядит следующим образом:

$$A = 1 \rightarrow R1$$
$$B = 1 \rightarrow R1$$

Следовательно, узел $R1$ рассылает в сети A и B вектор расстояний ($A = 1$, $B = 1$). Аналогично узел $R2$ рассылает в сети A, C, D вектор ($A = 1$, $C = 1$, $D = 1$). Узел $R1$ получает этот вектор из сети A, увеличивает расстояния на 1 ($A = 2$, $C = 2$, $D = 2$) и сравнивает с данными в своей таблице маршрутов. Новое расстояние до сети A оказывается больше, чем уже внесенное в таблицу ($A = 1$), следовательно, новое значение игнорируется. Поскольку сети C и D отсутствуют в его таблице маршрутов, они туда вносятся. В узле $R1$ имеем:

$$A = 1 \rightarrow R1$$
$$B = 1 \rightarrow R1$$
$$C = 2 \rightarrow R2$$
$$D = 2 \rightarrow R2$$

Узел $R4$ в свою очередь рассылает вектор расстояний ($D = 1$, $E = 1$) в сети D и E. Узел $R2$ получает этот вектор из сети D, увеличивает расстояния на 1, после чего добавляет себе в таблицу данные о сети E ($E = 2 \rightarrow R4$). Ранее из узла $R1$ он получил информацию о сети B и добавил себе в таблицу строку $B = 2 \rightarrow R1$. Узел $R2$ рассылает в сети A, C, D свой обновленный вектор расстояний ($A = 1$, $B = 2$, $C = 1$, $D = 1$, $E = 2$). Узел $R1$ получает этот вектор от $R2$ из сети A, увеличивает расстояния на 1: ($A = 2$, $B = 3$, $C = 2$, $D = 2$, $E = 3$) и замечает, что все указанные расстояния, кроме расстояния до сети E, больше либо равны значениям, имеющимся в его таблице. Сеть E в таблице узла $R1$ отсутствует, следовательно, она туда вносится. В результате в узле $R1$ таблица маршрутов имеет вид:

$$A = 1 \rightarrow R1$$
$$B = 1 \rightarrow R1$$
$$C = 2 \rightarrow R2$$
$$D = 2 \rightarrow R2$$

Далее маршрутизатор $R3$, ранее не работавший по каким-либо причинам, рассылает в сети B, C, E свой вектор ($B = 1$, $C = 1$, $E = 1$). Узел $R1$ получает этот вектор из сети B, увеличивает расстояния на 1 и обнаруживает, что расстояние $E = 2$ меньше имеющегося в таблице $E = 3$, следовательно запись о сети E в таблице заменяется на $E = 2 \rightarrow R3$. Остальные элементы полученного от $R3$ вектора не вызывают обновления таблицы. Итоговая таблица маршрутов маршрутизатора $R1$ выглядит следующим образом:

$$A = 1 \rightarrow R1$$
$$B = 1 \rightarrow R1$$
$$C = 2 \rightarrow R2$$
$$D = 2 \rightarrow R2$$
$$E = 2 \rightarrow R3$$

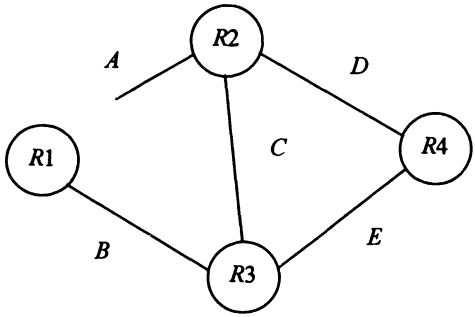


Рис. 5.42. Отказ порта сети *A* маршрутизатора *R*1

На этом алгоритм сходится, т. е. при неизменной топологии системы никакие векторы расстояний, получаемые маршрутизатором *R*1, больше не внесут изменений в таблицу маршрутов. Аналогичным образом алгоритм составления таблицы маршрутов работает и сходится на других маршрутизаторах. Для оперативного реагирования на внезапные изменения топологии сети векторы расстояний периодически широковещательно рассылаются каждым маршрутизатором. Очевидно, что вид построенной таблицы маршрутов может зависеть от порядка получения маршрутизатором векторов расстояний.

Изменение состояния RIP-системы. Рассмотрим случай, когда состояние системы неожиданно изменяется, например, маршрутизатор *R*1 отключается от сети *A* (рис. 5.42).

Узел *R*1 обнаруживает свое отсоединение от сети *A* и меняет таблицу маршрутов, устанавливая бесконечное расстояние до всех сетей, ранее достижимых через маршрутизаторы, подключенные к сети *A* (т. е. *R*2). В протоколе *RIP* значение бесконечности равно 16.

$$A = 16 \rightarrow R1$$
$$B = 1 \rightarrow R1$$
$$C = 16 \rightarrow R2$$
$$D = 16 \rightarrow R2$$
$$E = 2 \rightarrow R3$$

Вектор расстоянии, построенный на основании этой таблицы, рассылается в сеть *B*, чтобы маршрутизаторы, направлявшие свои данные через *R*1 в ставшие недоступными сети, если таковые маршрутизаторы существуют, соответственно изменили свои маршрутные таблицы. Допустим, в узле *R*3 имелась следующая таблица маршрутов:

$$A = 2 \rightarrow R2$$
$$B = 1 \rightarrow R3$$
$$C = 1 \rightarrow R3$$
$$D = 2 \rightarrow R4$$
$$E = 1 \rightarrow R3$$

Узел $R3$ периодически и широковещательно рассылает в сети B, C, E свой вектор расстояния ($A = 2$, $B = 1$, $C = 1$, $D = 2$, $E = 1$). Узел $R1$ получает этот вектор, увеличивает расстояния на 1: ($A = 3$, $B = 2$, $C = 2$, $D = 3$, $E = 2$) и замечает, что расстояния $A = 3$, $C = 2$ и $D = 3$ меньше бесконечности, следовательно, соответствующие записи таблицы маршрутов модифицируются и она принимает вид:

$$A = 3 \rightarrow R3$$
$$B = 1 \rightarrow R1$$
$$C = 2 \rightarrow R3$$
$$D = 3 \rightarrow R3$$
$$E = 2 \rightarrow R3$$

Таким образом, узел $R1$ построил маршруты в обход поврежденного участка и восстановил достижимость всех сетей.

К сожалению, поведение дистанционно-векторных протоколов при изменении топологии системы не всегда корректно и предсказуемо. Рассмотрим вышеописанную ситуацию в отношении протокола *RIP* с отсоединением узла $R1$ от сети A. Выше мы предполагали, что узел $R3$ не отправлял дейтаграмм через узел $R1$ (и, следовательно, изменение таблицы маршрутов в узле $R1$ не повлияло на таблицу узла $R3$). Предположим теперь, что $R3$ отправлял дейтаграммы в сеть A через $R1$, т. е. таблица в узле $R3$ имела вид:

$$A = 2 \rightarrow R1$$
$$B = 1 \rightarrow R3$$
$$C = 1 \rightarrow R3$$
$$D = 2 \rightarrow R4$$
$$E = 1 \rightarrow R3$$

После отсоединения $R1$ от сети A узел $R3$ получает от $R1$ вектор ($A = 16$, $B = 1$, $C = 16$, $D = 16$, $E = 2$). Проанализировав этот вектор, узел $R3$ делает вывод, что все указанные в нем расстояния больше значений, содержащихся в его маршрутной таблице, на основании чего этот вектор узлом $R3$ игнорируется. В свою очередь, узел $R3$ рассылает в сети B, C, E вектор ($A = 2$, $B = 1$, $C = 1$, $D = 2$, $E = 1$). Узел $R1$ получает этот вектор, увеличивает расстояния на 1: ($A = 3$, $B = 2$, $C = 2$, $D = 3$, $E = 2$) и замечает, что расстояния $A = 3$, $C = 2$ и $D = 3$ меньше бесконечности, следовательно, соответствующие записи таблицы маршрутов в узле $R1$ модифицируются и она принимает вид:

$$A = 3 \rightarrow R3$$
$$B = 1 \rightarrow R1$$
$$C = 2 \rightarrow R3$$
$$D = 3 \rightarrow R3$$
$$E = 2 \rightarrow R3$$

Очевидно, после этого содержимое таблиц узлов $R1$ и $R3$ стабилизируется. Рассмотрим теперь записи о достижении сети A в таблицах маршрутизаторов $R1$ и $R3$.

В узле $R1$:

$$A = 3 \rightarrow R3$$

В узле $R3$:

$$A = 2 \rightarrow R1$$

Таким образом, возникло зацикливание: данные, адресованные в сеть *A*, будут пересылаться между узлами *R*1 и *R*3 до тех пор, пока не истечет время жизни дейтаграмм и они не будут уничтожены.

Для того, чтобы избежать зацикливания, в алгоритм рассылки векторов расстояний вносятся дополнения.

1. Если дейтаграммы, адресованные в сеть X, посылаются через маршрутизатор G, находящийся в сети N, то в векторе расстояний, рассылаемом в сети N, расстояние до сети X не указывается.

В нашем примере узел *R*3 будет рассылать в сети *B* вектор ($B = 1, C = 1, D = 2, E = 1$). Элемент $A = 2$ не будет включен в этот вектор, потому что дейтаграммы в сеть *A* отправлены узлом *R*3 через узел *R*1, а узел *R*1 расположен в сети *B*. При рассылке узлом *R*3 вектора расстояний в другие сети элемент $A = 2$ будет указан (но не будут указаны какие-то другие элементы).

Модификация дополнения *1* позволяет ликвидировать более сложные особые ситуации, в том числе, некоторые случаи счета до бесконечности.

1A. Если дейтаграммы, адресованные в сеть X, посылаются через маршрутизатор G, находящийся в сети N, то в векторе расстояний, рассылаемом в сети N, расстояние до сети X полагается равным бесконечности. Тем не менее, и в этом случае могут возникать особые ситуации.

2. *Если маршрутизатор G объявляет новое расстояние до сети X, то это расстояние вносится в таблицы маршрутов узлов, отправляющих дейтаграммы в сеть X через G независимо от того, больше оно или меньше уже внесенного в таблицы расстояния.*

В нашем примере это означает, что если в маршрутной таблице узла *R*3 записано $A = 1 \rightarrow R1$ и *R*3 получает от *R*1 вектор с элементом $A = 16$, то несмотря на то, что $1 < \infty$, узел *R*3 модифицирует запись в таблице: $A = 16 \rightarrow R1$. Однако таким образом устраняются далеко не все случаи зацикливания.

Счет до бесконечности. При отказе оборудования может сложиться ситуация, при которой сеть, например *A*, оказывается изолированной, а маршрутизаторы, следуя алгоритму RIP будут обмениваться векторами до тех пор, пока расстояние до этой сети не станет равным бесконечности в маршрутных таблицах всех маршрутизаторов. В течение «счета до бесконечности» сеть *A* считается достижимой, поскольку расстояние до нее считается конечным и все дейтаграммы, адресованные в сеть *A*, отправляются маршрутизаторами согласно их таблицам по кругу.

Чтобы уменьшить отрицательный эффект этого явления, значение бесконечности не должно быть велико. В протоколе RIP оно равно 16, что, в свою очередь, ограничивает размер RIP-системы.

Работа протокола RIP. Каждому маршруту ставится в соответствие таймер тайм-аута и «сборщика мусора».Тайм-аут-таймер сбрасывается каждый раз, когда маршрут инициализируется или корректируется. Если со времени последней коррекции прошло 3 мин или получено сообщение о том, что вектор расстояния равен 16, маршрут считается закрытым. Но запись о нем не стирается до тех пор, пока не истечет время «уборки мусора» (2 мин).

При получении сообщения типа «ответ» для каждого содержащегося в нем элемента вектора расстояний модуль RIP выполняет следующие действия:

- проверяет корректность адреса сети и маски, указанных в сообщении;
- проверяет, не превышает ли метрика (расстояние до сети) бесконечности:
- некорректный элемент игнорируется;
- если метрика меньше бесконечности, она увеличивается на 1;
- производится поиск сети, указанной в рассматриваемом элементе вектора расстояний, в таблице маршрутов;
- если запись о такой сети в таблице маршрутов отсутствует и метрика в полученном элементе вектора меньше бесконечности, сеть вносится в таблицу маршрутов с указанной метрикой; в поле «Следующий маршрутизатор» заносится адрес маршрутизатора, приславшего сообщение; запускается таймер для этой записи в таблице;
- если искомая запись присутствует в таблице с метрикой больше, чем объявленная в полученном векторе, в таблицу вносятся новые метрика и, соответственно, адрес следующею маршрутизатора; таймер для этой записи перезапускается;
- если искомая запись присутствует в таблице и отправителем полученного вектора был маршрутизатор, указанный в поле «Следующий маршрутизатор» этой записи, то таймер для этой записи перезапускается; более того, если при этом метрика в таблице отличается от метрики в полученном векторе расстояний, в таблицу вносится значение метрики из полученного вектора;
- во всех прочих случаях рассматриваемый элемент вектора расстояний игнорируется.

Сообщения типа «ответ» модуль RIP рассылает каждые 30 с по широковещательному или групповому (только RIP-2) адресу. Рассылка «ответа» может происходить также вне графика, если была изменена маршрутная таблица. Стандарт требует, чтобы в этом случае «ответ» рассылался не немедленно после изменения таблицы маршрутов, а через случайный интервал длительностью от 1 до 5 с. Это позволяет несколько снизить нагрузку на сеть.

В каждую из сетей, подключенных к маршрутизатору, рассылается свой собственный вектор расстояний, построенный с учетом дополнения *1* (*1А*), сформулированного выше. Там, где это возможно, адреса сетей агрегируются (обобщаются), т. е. несколько подсетей с соседними адресами объединяются под одним, более общим адресом с соответствующим изменением маски.

В случае рассылки «ответа» вне графика (triggered response) посылается информация только о тех сетях, записи о которых были изменены. Информация о сетях с бесконечной метрикой посылается только в том случае, если она была недавно изменена.

При получении сообщения типа «запрос» с адресом 0.0.0.0 маршрутизатор рассылает в соответствующую сеть обычное сообщение типа ответ. При получении запроса с любым другим значением в поле (полях) «IP-адрес» посы-

лается ответ, содержащий информацию только о сетях, которые указаны. Такой ответ посылается только на адрес запросившего маршрутизатора (не широковещательно), при этом дополнение 1 (1А) не учитывается.

Конфигурирование RIP. Общий порядок действий при конфигурировании модуля RIP следующий:

- указать, какие сети, подключенные к маршрутизатору, будут включены в RIP-систему;
- указать nonbroadcast networks, т. е. сети со статической маршрутизацией (например, тупиковые сети, подсоединенные к внешнему миру через единственный шлюз), куда не нужно рассылать векторы расстояний;
- указать permanent routes – статические маршруты, например, маршрут по умолчанию за пределы автономной системы.

Протокол RIP очень прост, но так как он разрабатывался для локальных сетей, ему присущи следующие недостатки:

- малое значение бесконечности (из-за эффекта «счет до бесконечности») ограничивает размер RIP-системы четырнадцатью промежуточными маршрутизаторами в любом направлении. Кроме того, по той же причине весьма затруднительно использование сложных метрик, учитывающих не просто количество промежуточных маршрутизаторов, но и скорость и качество канала связи (чем медленнее канал, тем больше метрика);
- само явление счета до бесконечности вызывает сбои в маршрутизации;
- широковещательная рассылка векторов расстояний каждые 30 с ухудшает пропускную способность сети;
- время схождения алгоритма при создании маршрутных таблиц достаточно велико (по крайней мере, по сравнению с протоколами состояния связей);
- несмотря на то что каждый маршрутизатор начинает периодическую рассылку своих векторов, вообще говоря, в случайный момент времени (например, после включения), через некоторое время в системе наблюдается эффект синхронизации маршрутизаторов, сходный с эффектом синхронизации аплодисментов. Все маршрутизаторы рассылают свои вектора в один и тот же момент времени, что приводит к большим пикам трафика и отказам в маршрутизации дейтаграмм во время обработки большого количества одновременно полученных векторов;
- при использовании RIP таблица каждого маршрутизатора содержит полный список всех сетевых идентификаторов и возможных путей к ним. Она включает сотни или даже тысячи записей для большой объединенной IP-сети с многочисленными путями. Поскольку максимальный размер одного RIP-пакета составляет 512 байт, то для отправления больших таблиц маршрутизации необходимо множество RIP-пакетов.
- в таблице маршрутизации каждой записи о маршруте, полученном по RIP, назначен 3-минутный тайм-аут, по истечение которого не обновленные записи удаляются. Если маршрутизатор выходит из строя, распространение изменений по объединенной сети может занять несколько минут. Возникает проблема медленной конвергенции.

Протокол маршрутизации OSPF

Протокол маршрутизации OSPF (Open Shortest Path First) представляет собой протокол состояния связей, использующий алгоритм SPF поиска кратчайшего пути в графе. OSPF применяют для внутренней маршрутизации в системах сетей любой сложности. Рассмотрим работу алгоритма SPF и построение маршрутов на примере OSPF-системы, состоящей из маршрутизаторов, соединенных линиями связи типа «точка-точка» (рис. 5.43).

Метрика представляет собой оценку качества связи в данной сети (на данном физическом канале); чем меньше метрика, тем лучше качество соединения. Метрика маршрута равна сумме метрик всех связей (сетей), входящих в маршрут. В простейшем случае, как это имеет место в протоколе RIP, метрика каждой сети равна единице, а метрика маршрута равна его длине в хопах.

Поскольку при работе алгоритма SPF ситуации, приводящие к счету до бесконечности, отсутствуют, значения метрик могут варьироваться в широком диапазоне. Кроме того, протокол OSPF позволяет определить для любой сети различные значения метрик в зависимости от типа сервиса (тип сервиса запрашивается дейтаграммой в соответствии со значением поля «Тип обслуживания» (ToS) ее заголовка.) Для каждого типа сервиса вычисляется свой маршрут, и дейтаграммы, затребовавшие наиболее скоростной канал, могут быть отправлены по одному маршруту, а затребовавшие наименее дорогостоящий канал – по другому.

Метрика сети, оценивающая пропускную способность, определяется как количество секунд, требуемое для передачи 100 Мбит через физическую среду данной сети. Например, метрика сети на базе 10Base-T Ethernet равна 10, а метрика выделенной линии 56 кбит/с – 1785. Метрика канала со скоростью передачи данных 100 М бит/с и выше равна единице.

Порядок расчета метрик, оценивающих надежность, задержку и стоимость, не определен. Администратор, желающий поддерживать маршрутизацию по этим типам сервисов, должен сам назначить разумные и согласованные метрики по этим параметрам.

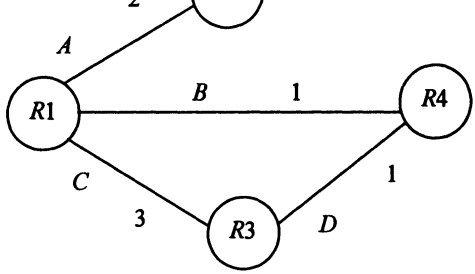


Рис. 5.43. Пример структуры OSPF-системы:
*R*1, *R*2, *R*3, *R*4 – маршрутизаторы; *A, B, C, D* – связи,
1 – 3 – метрика каждой связи

От	До	Сеть	Метрика
*R*1	*R*2	*A*	2
*R*1	*R*3	*C*	3
*R*1	*R*4	*B*	1
*R*2	*R*1	*A*	2
*R*3	*R*1	*C*	3
*R*3	*R*4	*D*	1
*R*4	*R*1	*B*	1
*R*4	*R*3	*D*	1

Рис. 5.44. База данных состояния связей

Если не требуется маршрутизация с учетом типа сервиса (или маршрутизатор ее не поддерживает), используют метрику по умолчанию, равную метрике по пропускной способности. Именно ее и будем использовать в дальнейшем.

Для работы алгоритма SPF на каждом маршрутизаторе строится база данных состояния связей, представляющая собой полное описание графа OSPF-системы. При этом вершинами графа являются маршрутизаторы, а ребрами – соединяющие их связи. Базы данных на всех маршрутизаторах идентичны. За создание баз данных и поддержку их взаимной синхронизации при изменениях в структуре системы сетей отвечают другие алгоритмы, содержащиеся в протоколе OSPF. Рассмотрим эти алгоритмы позже, а сейчас будем считать, что базы данных на всех маршрутизаторах каким-то образом построены, синхронизированы и правильно описывают граф системы в данный момент времени.

База данных состояния связей (рис. 5.44) представляет собой таблицу, где для каждой пары смежных вершин графа (маршрутизаторов) указано ребро (связь), их соединяющее, и метрика этого ребра. Граф считается ориентированным, т. е. ребро, соединяющее вершину *R*1 с вершиной *R*2, и ребро, соединяющее вершину *R*2 с вершиной *R*1, могут быть различны или это может быть одно и то же ребро, но с разными метриками.

Алгоритм поиска кратчайшего пути. Рассмотрим алгоритм SPF поиска кратчайшего пути, предложенный Е.В. Дейкстрой (E.W. Dijkstra). Алгоритм SPF, основываясь на базе данных состояния связей, вычисляет кратчайшие пути между заданной вершиной S-графа и всеми остальными вершинами. Результатом работы алгоритма является таблица, где для каждой вершины V-графа указан список ребер, соединяющих заданную вершину S с вершиной V по кратчайшему пути.

Введем следующие обозначения:

E – множество обработанных вершин, т. е. вершин, кратчайший путь к которым от заданной вершины S уже найден;

R – множество оставшихся вершин графа (т. е. множество вершин графа за вычетом множества E);

O – упорядоченный список путей.

Шаг 1. Инициализировать E={S}, R={все вершины графа, кроме S}. Поместить в O все односегментные (длиной в одно ребро) пути, начинающиеся из S, отсортировав их в порядке возрастания метрик.

Шаг 2. Если O пуст или первый путь в O имеет бесконечную метрику, то отметить все вершины в R как недостижимые и закончить работу алгоритма.

Шаг 3. Рассмотрим P – кратчайший путь в списке O. Удалить P из O. Пусть V – последний узел в P. Если V ∈ E, перейти на шаг 2; иначе P является кратчайшим путем из S в V (будем записывать как V:P); перенести V из R в E.

Шаг 4. Построить набор новых путей, подлежащих рассмотрению, путем добавления к пути P всех односегментных путей, начинающихся из V. Метрика каждого нового пути равна сумме метрики P и метрики соответствующего односегментного отрезка, начинающегося из V. Добавить новые пути в упорядоченный список O, поместив их на места в соответствии со значениями метрик. Перейти к шагу 2.

Все вычисления вычисляются локально по известной базе данных, а потому значительно быстрее по сравнению с дистанционно-векторными протоколами, при этом результаты получаются на основе полной, а не частичной информации о графе системы сетей.

Предположим, что к маршрутизатору *R*4 подключена сеть *N*1 компьютеров (хостов) $H_1 - H_k$. Следуя разобранной выше модели, каждый хост должен быть также вершиной графа OSPF-системы, хотя сам и не строит базу данных и не производит вычисления маршрутов. Тем не менее, информация о связях маршрутизатора *R*4 с каждым из хостов сети *N*1 и о метриках этих связей должна быть внесена в базу данных, чтобы все остальные маршрутизаторы системы могли построить маршруты от себя до этих хостов. Очевидно, что такая процедура неэффективна. Вместо информации о связях с каждым хостом в базу данных вносится информация о связи с сетью, т. е. сама IP-сеть становится вершиной графа системы, соединенной с маршрутизатором *R*4 некоторой связью P (рис. 5.45).

В данном случае сеть, точнее ее адрес, используется как *обобщающий* идентификатор группы *хостов*, находящихся в одной IP-сети, к которой маршрутизатор *R*4 непосредственно подключен. Вершина *N*1 называется *тупиковой сетью* (*stub network*); все узлы сети, обозначаемые этой вершиной, являются хостами, у которых установлен маршрут по умолчанию, направленный на маршрутизатор *R*4.

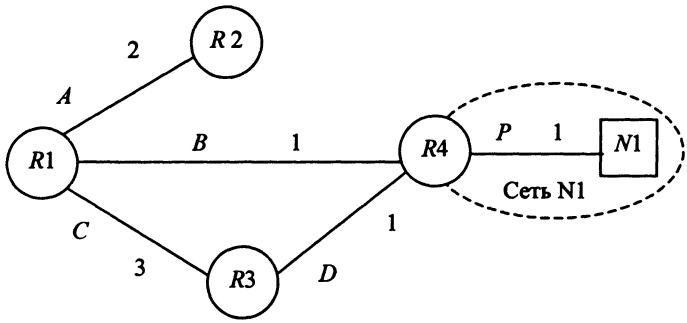


Рис. 5.45. Дополнение IP-сети в OSPF-систему

От	До	Сеть	Метрика
*R*4	*N*1	*P*	1

Рис. 5.46. Запись в базе данных для тупиковой сети

Протокол OSPF проводит разграничение хостов и маршрутизаторов. Если к IP-сети *N*1 подключен еще и один из интерфейсов маршрутизатора *R*2, то связь между *R*2 и *R*4 будет установлена отдельно, как если бы они были соединены двухточечной линией связи (при этом у маршрутизатора *R*2 также будет связь с тупиковой сетью *N*1).

При подключении тупиковой сети *N*1 в базе данных состояния связей появится запись (рис. 5.46).

Связей, направленных из вершины *N*1, в базе данных нет и не будет, потому что вершина *N*1 не является маршрутизатором. Построение маршрутов до вершины *N*1 (т. е. фактически сразу до всех хостов сети *N*1) будет осуществлено каждым маршрутизатором обычным способом по алгоритму SPF.

Поддержка множественных маршрутов. Если между двумя узлами сети существует несколько маршрутов с одинаковыми или близкими по значению метриками, протокол OSPF позволяет направлять части трафика по этим маршрутам в пропорции, соответствующей значениям метрик. Например, если есть два альтернативных маршрута с метриками 1 и 2, то две трети трафика будет направлено по первому из них, а оставшаяся треть – по второму. Положительный эффект такого механизма заключается в уменьшении средней задержки прохождения дейтаграмм между отправителем и получателем, а также в уменьшении колебаний значения средней задержки.

Менее очевидное преимущество поддержки множественных маршрутов состоит в следующем. Если при использовании только одного из возможных маршрутов этот маршрут внезапно выходит из строя, весь трафик будет разом перемаршрутизирован на альтернативный маршрут, при этом во время процесса массового переключения больших объемов трафика с одного маршрута на другой весьма велика вероятность образования затора на новом маршруте. Если же до аварии использовалось разделение трафика по нескольким маршрутам, отказ одного из них вызовет перемаршрутизацию лишь части трафика, что существенно сгладит нежелательные эффекты.

Рассмотрим следующий пример (рис. 5.47). Узел *R*1 отправляет данные в узел *R*3, используя поддержку множественных маршрутов, по маршрутам *C* (2/3 трафика) и *AB* (1/3 трафика). Однако узел *R*2 тоже поддерживает механизм множественных путей, и когда к нему пребывают дейтаграммы, адресованные в *R*3 (в том числе, и отправленные из *R*1), он применяет к ним ту же логику, т. е. 2/3 из них отправляет в *R*3 по маршруту *B*, а одну треть – по маршруту *AC*. Следовательно, 1/9 дейтаграмм, отправленных узлом *R*1 в узел *R*3, возвращаются опять в узел *R*1, который 1/3 из них опять отправляет в *R*3 по маршруту *C*, а 2/3 – по маршруту *AB* через

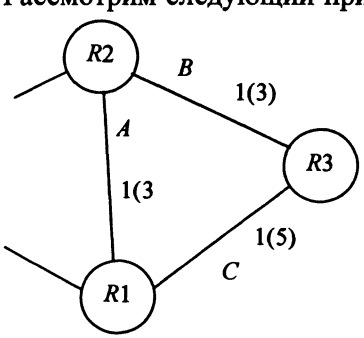


Рис. 5.47. Частичное зацикливание дейтаграмм

узел *R*2 и т. д. В итоге сформировался «частичный цикл» при посылке дейтаграмм из *R*1 в *R*3, который, помимо частичного зацикливания дейтаграмм, ведет к быстрой перегрузке линии *A*.

Избежать зацикливание дейтаграмм позволяет следующее правило:

если узел X отправляет данные в узел Y, он может пересылать их через узел Q только в том случае, если Q ближе к Y, чем к X.

В приведенном примере, следуя этому правилу, *R*1 не может посылать данные в *R*3 через *R*2, поскольку *R*2 не ближе к *R*3, чем *R*1. Однако такая посылка возможна, если связи между узлами имеют соответствующие метрики, на рис.5.47 эти значения приведены в скобках.

Для реализации построения дополнительных альтернативных маршрутов с учетом вышеприведенного правила в алгоритме *SPF* необходимо внести изменения в шаг 3 и добавить шаг 3*а*. Ниже приведена новая версия алгоритма SPF, в которой изменение и дополнение показаны курсивом.

Алгоритм SPF с поддержкой множественных маршрутов.

Шаг 1. Инициализировать E={S}, R={все вершины графа, кроме S}. Поместить в O все односегментные (длиной в одно ребро) пути, начинающиеся из вершины S, отсортировав их в порядке возрастания метрик.

Шаг 2. Если O пуст или первый путь в O имеет бесконечную метрику, то отметить все вершины в R как недостижимые и закончить работу алгоритма.

Шаг 3. Рассмотрим P – кратчайший путь в списке O. Удалить P из O. Пусть V – последний узел в P. Если V принадлежит E, *перейти на шаг 3а*; иначе P является кратчайшим путем из S в V; перенести V из R в E. *Перейти на шаг 4.*

Шаг 3а. Рассмотрим W, узел, предшествующий V в пути P. Если расстояние от S до W меньше расстояния от S до V, то обозначим P как приемлемый альтернативный путь к V. В любом случае перейти на шаг 2.

Шаг 4. Построить набор новых путей, подлежащих рассмотрению, путем добавления к пути P всех односегментных путей, начинающихся из узла V. Метрика каждого нового пути равна сумме метрики P и метрики соответствующего односегментного отрезка, начинающегося из V. Добавить новые пути в упорядоченный список O, поместив их на места в соответствии со значениями метрик. Перейти на шаг 2.

Накладывающиеся маршруты. Пусть в графе OSPF-системы некий маршрутизатор имеет связи с вершинами N и M, которые представляют сети хостов, подключенные к различным интерфейсам маршрутизатора. Это означает, что в таблице маршрутов этого маршрутизатора, будет две записи: для сети N и для сети M.

Предположим теперь, что адрес и маска сети M таковы, что она является подсетью сети N. Например, IP-адрес сети N равен 172.16.0.0, а маска сети – 255.255.0.0; для сети M –172.16.5.0 и 255.255.255.0 соответственно.

В этом случае дейтаграммы, следующие по адресу, находящемуся в обеих сетях, будут отправлены в сеть с более длинной маской. Например, адрес 172.16.5.1 находится как в сети N, так и в сети M, но маска сети M длиннее, следовательно, дейтаграмма, следующая по этому адресу, будет отправлена в сеть M.

Внешние маршруты. Для достижения сетей, не входящих в OSPF-систему (в автономную систему), используют *пограничные маршрутизаторы автономной системы* (ASBR – Autonomous System Border Router), имеющие связи, уходящие за пределы системы.

ASBR вносят в базу данных состояния связей данные о сетях за пределами системы, достижимых через тот или иной маршрутизатор ASBR. Такие сети, а также ведущие к ним маршруты называются *внешними* (external).

В простейшем случае, если в системе есть только один *ASBR*, он объявляет через себя маршрут по умолчанию (default route) и все дейтаграммы, адресованные в сети, не входящие в базу данных системы, отправляются через этот маршрутизатор. Если в системе несколько ASBR, то, возможно, внутренним маршрутизаторам системы придется выбирать, через какой именно пограничный маршрутизатор нужно отправлять дейтаграммы в ту или иную внешнюю сеть. Это делается на основе специальных записей, вносимых ASBR в базу данных системы. Такие записи содержат адрес и маску внешней сети и метрику расстояния до нее, которая может быть сравнимой с метриками, используемыми в OSPF-системе. Если возможно, адреса нескольких внешних сетей *агрегируются* в общий адрес с более короткой маской.

Все или некоторые внешние маршруты могут быть сконфигурированы администратором (в том числе единственный маршрут по умолчанию) либо ASBR может получать информацию о внешних маршрутах от протоколов внешней маршрутизации.

Построение базы данных состояния связей. *Протокол Hello.* После инициализации модуля OSPF (например, после подачи питания на маршрутизатор) через все интерфейсы, включенные в OSPF-систему, начинают рассылаться Hello-сообщения. Задача Hello-протокола состоит в обнаружении соседей и установлении с ними отношений *смежности*. *Соседями* называют OSPF-маршрутизаторы, подключенные к одной сети (к одной линии связи) и обменивающиеся Hello-сообщениями. *Смежными* называют *соседние* OSPF-маршрутизаторы, которые приняли решение обмениваться друг с другом информацией, необходимой для синхронизации базы данных состояния связей и построения маршрутов. Не все соседи становятся смежными.

Другой задачей протокола Hello является выбор выделенного маршрутизатора в сети с множественным доступом, к которой подключено несколько маршрутизаторов.

Hello-пакеты периодически рассылаются и после того, как соседи обнаружены. Так маршрутизатор контролирует состояние своих связей с соседями и может своевременно обнаружить изменение этого состояния (например, обрыв связи или отключение одного из соседей). Обрыв связи можно также об-

наружить и с помощью протокола канального уровня, который просигнализирует о недоступности канала.

В сетях с возможностью широковещательной рассылки (broadcast networks) Hello-пакеты рассылаются по мультикастинговому адресу 224.0.0.5 («Всем OSPF-маршрутизаторам»). В других сетях все возможные адреса соседей должны быть введены администратором.

Протокол обмена. После установления отношений смежности для каждой пары смежных маршрутизаторов осуществляется синхронизация их баз данных. Эта же операция проводится при восстановлении ранее разорванного соединения, поскольку в образовавшихся после аварии двух изолированных подсистемах базы данных развивались независимо друг от друга. Синхронизация баз данных происходит с помощью *протокола обмена* (Exchange protocol).

Сначала маршрутизаторы обмениваются только описаниями своих баз данных (Database DescrIPtion), содержащими идентификаторы записей и номера их версий, что позволяет избежать пересылки всего содержимого базы данных, если нужно синхронизировать только несколько.

Во время этого обмена каждый маршрутизатор формирует список записей, содержимое которых он должен запросить (т. е. эти записи в его базе данных устарели либо отсутствуют), и соответственно отправляет пакеты запросов о состоянии связей (Link State Request). В ответ он получает содержимое последних версий нужных ему записей в пакетах типа «Обновление состояния связей (Link State Update)».

После синхронизации баз данных осуществляется построение маршрутов, как описано ранее.

Протокол затопления. Каждый маршрутизатор отвечает за те и только те записи в базе данных состояния связей, которые описывают связи, *исходящие* от данного маршрутизатора. Это означает, что при образовании новой связи, изменении в состоянии связи или ее исчезновении (обрыве), маршрутизатор, ответственный за эту связь, должен соответственно изменить свою копию базы данных и немедленно известить все остальные маршрутизаторы *OSPF*-системы о произошедших изменениях, чтобы они также внесли исправления в свои копии базы данных.

Подпротокол OSPF, выполняющий эту задачу, *называется протоколом затопления* (Flooding protocol). Этот протокол пересылает сообщения типа «Обновление состояния связей (Link State Update)», получение которых подтверждают сообщения типа «Link State Acknowledgment».

Каждая запись о состоянии связей имеет свой номер (номер версии), который также хранится в базе данных. Каждая новая версия записи имеет бóльший номер. При рассылке сообщений об обновлении записи в базе данных номер записи также включается в сообщение для предотвращения попадания в базу данных устаревших версий.

Маршрутизатор, ответственный за запись об изменившейся связи, рассылает сообщение «Обновление состояния связи» по всем интерфейсам. Однако новые версии состояния одной и той же связи должны появляться не чаще, чем оговорено определенной константой. Далее на всех маршрутизаторах OSPF-системы действует следующий алгоритм.

1. Получить сообщение. Найти соответствующую запись в базе данных.
2. Если запись не найдена, добавить ее в базу данных, передать сообщение по всем интерфейсам.
3. Если номер записи в базе данных меньше номера пришедшего сообщения, заменить запись в базе данных, передать сообщение по всем интерфейсам.
4. Если номер записи в базе данных больше номера пришедшего сообщения и эта запись не была недавно разослана, разослать содержимое записи из базы данных через тот интерфейс, откуда пришло сообщение. Понятие «недавно» определяется значением константы.
5. В случае равных номеров записей сообщение игнорировать.

Протокол OSPF устанавливает для записи характеристику возраст. Возраст равен нулю при создании записи в базе данных. При затоплении OSPF-системы сообщениями с данной записью каждый маршрутизатор, который ретранслирует сообщение, увеличивает возраст записи на определенную величину. Кроме этого, возраст увеличивается на единицу каждую секунду. Из-за разницы во времени пересылки, в количестве промежуточных маршрутизаторов и по другим причинам возраст одной и той же записи в базах данных на разных маршрутизаторах может несколько различаться.

При достижении возрастом максимального значения (60 мин), соответствующая запись расценивается маршрутизатором как просроченная и непригодная для вычисления маршрутов. Такая запись должна быть удалена из базы данных.

Поскольку базы данных на всех маршрутизаторах системы должны быть идентичны, просроченная запись должна быть удалена из всех копий базы данных на всех маршрутизаторах. Это осуществляется с использованием протокола затопления: маршрутизатор затапливает систему сообщением с просроченной записью. Соответственно, в описанный выше алгоритм обработки сообщения вносятся дополнения, связанные с получением просроченного сообщения и удалением соответствующей записи из базы данных.

Чтобы записи в базе данных не устаревали, маршрутизаторы, ответственные за них, должны через каждые 30 мин затапливать систему сообщениями об обновлении записей, даже если состояние связей не изменилось. Содержимое записей в этих сообщениях неизменно, но номер версии больше, а возраст равен нулю.

Вышеописанные протоколы обеспечивают актуальность информации, содержащейся в базе данных состояния связей, оперативное реагирование на изменения в топологии системы сетей и синхронизацию копий базы данных на всех маршрутизаторах системы. Для обеспечения надежности передачи данных реализован механизм подтверждения приема сообщений и вычисляется контрольная сумма. В протоколе OSPF может быть применена аутентификация сообщений.

5.6. Протоколы II уровня стека TCP/IP

Протокол управления передачей TCP

Основные характеристики и понятия протокола. Протокол управления передачей TCP (Transmission Control Protocol) является протоколом транспортного уровня и базируется на возможностях, предоставляемых межсетевым протоколом IP. Основная задача TCP – обеспечение надежной передачи данных в сети. Его транспортный адрес в заголовке IP-сегмента равен 6. Описание протокола TCP дано в RFC 793.

Основные характеристики протокола TCP следующие:

- реализует взаимодействие в режиме с установлением логического (виртуального) соединения;
- обеспечивает двунаправленную дуплексную связь;
- организует потоковый (с точки зрения пользователя) тип передачи данных;
- дает возможность пересылки части данных как «экстренных»;
- для идентификации партнеров по взаимодействию на транспортном уровне использует 16-битовые «номера портов»;
- реализует принцип «скользящего окна» (sliding window) для повышения скорости передачи;
- поддерживает ряд механизмов для обеспечения надежной передачи данных.

В то время как задачей сетевого уровня является передача данных между произвольными узлами сети, задача транспортного уровня заключается в передаче данных между любыми *прикладными процессами*, выполняющимися на любых узлах сети. Действительно, после того как пакет средствами протокола IP доставлен в компьютер-получатель, данные необходимо направить конкретному процессу-получателю. Каждый компьютер может выполнять несколько процессов, более того, прикладной процесс тоже может иметь несколько точек входа, выступающих в качестве адреса назначения для пакетов данных.

Пакеты, поступающие на транспортный уровень, организуются операционной системой в виде множества очередей к точкам входа различных прикладных процессов. В терминологии TCP/IP такие системные очереди называют *портами*. Таким образом, адресом назначения, используемым на транспортном уровне, является идентификатор (номер) порта прикладного сервиса. Номер порта, задаваемый транспортным уровнем, в совокупности с номером сети и номером компьютера, задаваемыми сетевым уровнем, однозначно определяют прикладной процесс в сети.

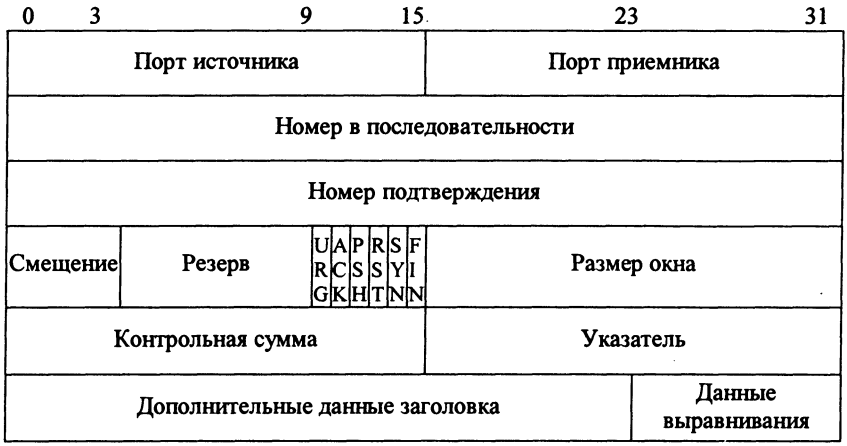


Рис. 5.48. Формат заголовка пакета TCP

Несмотря на то что для пользователя передача данных с использованием протокола TCP выглядит как потоковая, на самом же деле обмен между партнерами осуществляется посредством пакетов данных, которые мы будем называть «TCP-пакетами».

Формат заголовка пакета TCP и назначение полей. На рис. 5.48 приведен формат заголовка TCP-пакета. Порт источника и порт приемника представляют собой 16-битовые поля, содержащие номера портов, соответственно, источника и адресата TCP-пакета. Номер в последовательности (sequence number) – 32-битовое поле, содержимое которого определяет (косвенно) положение данных TCP-пакета внутри исходящего потока данных, существующего в рамках текущего логического соединения.

В момент установления логического соединения каждый из двух партнеров генерирует свой начальный «номер в последовательности», основное требование к которому – не повторяться в промежутке времени, в течение которого TCP-пакет может находиться в сети (по сути, это время жизни IP-сегмента). Партнеры обмениваются этими начальными номерами и подтверждают их получение. Во время отправления TCP-пакетов с данными поле «номер в последовательности» содержит сумму начального номера и количества байт ранее переданных данных.

Номер подтверждения (acknowledgement number) – 32-битовое поле, содержимое которого определяет (косвенно) количество принятых данных из входящего потока к TCP-модулю, формирующему TCP-пакет.

Смещение данных – 4-битовое поле, содержащее длину заголовка TCP-пакета в 32-битовых словах и используемое для определения начала расположения данных в TCP-пакете.

Флаг URG – бит, единичное значение которого означает, что TCP-пакет содержит важные (urgent) данные. Подробно информация содержится в поле данных TCP-пакета, определенного как «Важные данные».

Флаг ACK – бит, единичное значение которого означает, что TCP-пакет содержит в поле «номер подтверждения» верные данные.

Флаг PSH – бит, единичное значение которого означает, что данные, содержащиеся в TCP-пакете, должны быть немедленно переданы прикладной программе, для которой они адресованы. Подтверждение для TCP-пакета, содержащего единичное значение во флаге PSH, означает, что и все предыдущие TCP-пакеты достигли адресата.

Флаг RST – бит, устанавливается в единицу в TCP-пакете, отправляемом в ответ на получение неверного TCP-пакета. Также может означать запрос на переустановление логического соединения.

Флаг SYN – бит, единичное значение которого означает, что TCP-пакет представляет собой запрос на установление логического соединения. Получение пакета с установленным флагом SYN должно быть подтверждено принимающей стороной.

Флаг FIN – бит, единичное значение которого означает, что TCP-пакет представляет собой запрос на закрытие логического соединения и является признаком конца потока данных, передаваемых в этом направлении. Получение пакета с установленным флагом FIN должно быть подтверждено принимающей стороной.

Размер окна – 16-битовое поле, содержащее количество байт информации, которое может принять в свои внутренние буфера TCP-модуль, отправляющий партнеру данный TCP-пакет. Данное поле используется принимающим поток данных TCP-модулем для управления интенсивностью этого потока, так, установив нулевым значение этого поля, можно полностью остановить передачу данных, которая будет возобновлена только, когда размер окна примет достаточно большое значение. Максимальный размер окна зависит от реализации, в некоторых реализациях максимальный размер устанавливается системным администратором (типичное значение максимального размера окна – 4096 байт). Определение оптимального размера окна – одна из наиболее сложных задач реализации протокола TCP.

Контрольная сумма – 16-битовое поле, содержащее контрольную сумму, подсчитанную для TCP-заголовка, данных пакета и ряда полей IP-заголовка.

Указатель – 16-битовое поле, содержащее указатель (в виде смещения) на первый байт в теле TCP-пакета, начинающий последовательность важных (urgent) данных.

Дополнительные данные заголовка – последовательность полей произвольной длины, описывающих необязательные данные заголовка. Протокол TCP определяет только три типа дополнительных данных заголовка:

- конец списка полей дополнительных данных;
- пусто (No Operation);
- максимальный размер пакета.

Дополнительные данные последнего типа посылаются в TCP-заголовке в момент установления логического соединения для выражения готовности TCP-модулем принимать пакеты длиннее 536 байт. В UNIX-реализациях длина пакета обычно определяется максимальной длиной IP-сегмента для сети.

Номера портов играют роль адресов транспортного уровня, идентифицируя на конкретных узлах сети, по сути дела, потребителей транспортных услуг, предоставляемых как протоколом TCP, так и протоколом UDP. При этом протоколы TCP и UDP имеют свои собственные адресные пространства: например, порт номер 513 для TCP не идентичен порту номер 513 для UDP.

Примечание. Своя собственная адресация на транспортном уровне стека протоколов сетевого взаимодействия необходима для обеспечения возможности функционирования на узле сети одновременно многих сетевых приложений. Наличие в TCP-заголовке номера порта позволяет TCP-модулю, получающему последовательности TCP-пакетов, формировать раздельные потоки данных к прикладным программам.

Взаимодействие прикладных программ, использующих транспортные услуги протокола TCP (или UDP), строится согласно модели «клиент-сервер», которая подразумевает, что одна программа (сервер) всегда пассивно ожидает обращения к ней другой программы (клиента). Связь программы-клиента и сервера идентифицируется пятеркой:

1. Используемый транспортный протокол (TCP или UDP);
2. IP-адрес сервера;
3. Номер порта сервера;
4. IP-адрес клиента;
5. Номер порта клиента.

Для того, чтобы клиент мог обращаться к необходимому ему серверу, он должен знать номер порта, по которому сервер ожидает обращения к нему («слушает сеть»). Локальное присвоение номера порта заключается в том, что разработчик некоторого приложения просто связывает с ним любой доступный, произвольно выбранный числовой идентификатор, обращая внимание на то, чтобы он не входил в число зарезервированных номеров портов. В дальнейшем все удаленные запросы к данному приложению от других приложений должны адресоваться с указанием назначенного ему номера порта.

Как должны назначаться номера протокольных портов? Эта проблема важна, так как два компьютера должны договариваться о номерах портов, прежде чем они смогут взаимодействовать. Например, когда компьютер А хочет получить файл от компьютера В, он должен знать, какой порт в компьютере В использует программа передачи файла. Существуют два фундаментальных подхода к назначению портов.

Первый подход использует централизованное управление назначением. Все договариваются позволить центральному органу назначать номера всем необходимым портам и затем опубликовать список назначений. Тогда все программы создаются в соответствии с этим списком. Этот подход иногда называют «универсальным назначением». Такие назначения портов становятся широко известными назначениями, а номера портов фиксированы и носят название «хо-

шо известных номеров портов» (well-known port numbers). Централизованное присвоение сервисам номеров портов выполняется организацией Internet Assigned Numbers Authority.

Второй подход использует динамическое назначение. При этом подходе номера портов неизвестны всем. Вместо этого само сетевое обеспечение назначает порт, когда программа в этом нуждается. Чтобы узнать о текущем назначении портов на другом компьютере, нужно послать запрос, в котором задан примерно такой вопрос: «как мне вызвать службу передачи файлов?». Компьютер-получатель ответит, какой порт необходимо использовать.

Разработчики TCP/IP приняли смешанный подход, при котором группа портов назначается априорно, но большинство из них можно свободно использовать для любых целей прикладными программами в локальной сети. Централизованно назначаемые номера портов начинаются с маленьких значений и затем увеличиваются, а порты с большими значениями используются для динамического назначения.

На рис. 5.49 приведены примеры номеров портов для некоторых служб.

В протоколе TCP использован принцип «скользящего окна», который обеспечивает «опережающую» посылку данных с «отложенным» их подтверждением. Следует отметить недостаток этого механизма: если в течение некоторого времени не будет получено «отсроченное» подтверждение ранее отправленного пакета, то отправляющий TCP-модуль будет вынужден повторить посылку всех TCP-пакетов, начиная с неподтвержденного. Размер окна, как правило, определяется объемом свободного места в буферах принимающего TCP-модуля.

Протокол TCP предусматривает возможность информирования принимающей стороны взаимодействия отправляющей стороной о наличии в TCP-пакете важных данных (urgent data), требующих особого внимания согласно логике прикладной задачи. Отличие важных данных от данных основного потока заключается в том, что принимающая сторона должна, как правило, обработать их прежде ранее полученных, но еще не обработанных данных потока.

Для индикации наличия в TCP-пакете важных данных используется флаг URG TCP-заголовка, местоположение важных данных в теле TCP-пакета определяется полем «Указатель» TCP-заголовка – оно задает смещение первого байта важных данных в теле TCP-пакета.

Служба	Номер порта	Протокол
ftp-data	20	TCP
ftp	21	TCP
telnet	23	TCP
smtp	25	TCP
time	37	TCP
time	37	UDP
finger	79	TCP
portmap	111	TCP
portmap	111	UDP
exec	512	TCP
login	513	TCP
who	513	UDP
shell	514	TCP
talk	517	UDP
route	520	UDP
Xserver	6000	TCP

Рис. 5.49. Номера портов для некоторых служб

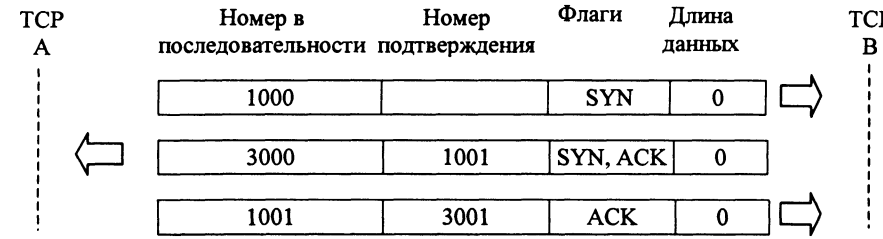


Рис. 5.50. Установление логического соединения

Протокол TCP предусматривает передачу важных (urgent) данных в рамках общего потока данных («in-band»). Существуют протоколы (например ISO), поддерживающие режим передачи важных (expedited) данных вне общего потока данных («out-band»), что в общем случае быстрее.

Этапы TCP-взаимодействия. Взаимодействие партнеров с использованием протокола TCP строится в три этапа:

- установление логического соединения;
- обмен данными;
- закрытие соединения.

Рис. 5.50–5.52 иллюстрируют последовательность обмена TCP-пакетами двумя TCP-модулями: A и B. TCP-пакеты представлены тремя полями TCP-заголовка («Номер в последовательности», «Номер подтверждения», «Флаги») и числом, характеризующим длину данных из которых тело TCP-пакета (заметим, что реально поля «Длина данных» в TCP-заголовке нет). Стрелками показаны направления пересылки пакетов.

Рис. 5.50 демонстрирует этап установления соединения, реализуемый как «трехшаговое рукопожатие» (three-way handshake). На первом шаге TCP-модуль A, играя роль клиента, посылает TCP-модулю B пакет с установленным флагом SYN и начальным значением номера в последовательности, равным 1000. TCP-модуль B, будучи готов со своей стороны установить соединение, отвечает TCP-пакетом, подтверждающим правильный прием запроса (поле «Номер подтверждения» на 1 больше начального номера в последовательности для TCP-модуля A и среди флагов есть установленный в 1 флаг ACK) и информирующим о готовности установить соединение (установлен флаг SYN и установлено значение 3000 начального номера последовательности). На третьем шаге TCP-модуль A подтверждает правильность приема TCP-пакета от B.

На рис. 5.51 показан этап двустороннего обмена данными между TCP-модулями A и B. TCP-модуль, принимающий адресованные ему данные, всегда подтверждает их прием, вычисляя значение поля «Номер подтверждения» в заголовке ответного TCP-пакета как сумму пришедшего «Номера в последовательности» и длины правильно принятых данных. Посылка данных к партнеру и подтверждение принятых от него данных реализуются в рамках одного TCP-пакета.

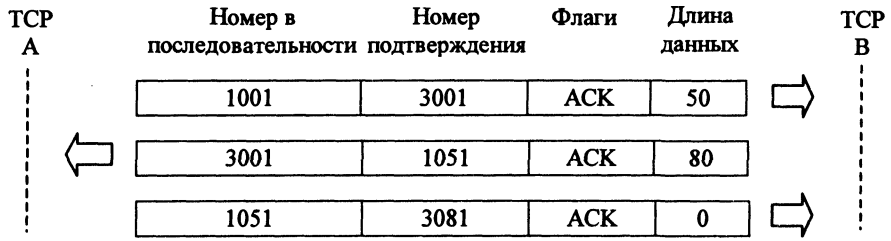


Рис. 5.51. Обмен данными

Рис. 5.52 иллюстрирует закрытие соединения по инициативе TCP-модуля A, посылающего партнеру TCP-пакет с установленным флагом FIN. Прием запроса на закрытие соединения TCP-модуль B подтверждает пакетом, содержащем в своем заголовке поле «Номер подтверждения», значение которого (1052) на 1 больше значения принятого «Номера в последовательности» (1051). После этого посылка каких-либо данных TCP-модулем A невозможна, однако модуль B имеет данные для передачи, которые он отправляет TCP-модулю A и получает подтверждение на их прием. Затем TCP-модуль B формирует пакет с флагом FIN, после подтверждения его приема соединение считается закрытым.

При подтверждении TCP-пакетов, содержащих единичные флаги SYN или FIN, значение поля «Номер подтверждения» на 1 больше значения соответствующего поля «Номер в последовательности», несмотря на то, что никакие данные в подтверждаемых TCP-пакетах не передаются.

Примечание. Приведенный пример не рассматривает ситуации, связанные с «потерей» TCP-пакетов в сети, и их обработку, связанную с повторной передачей данных.

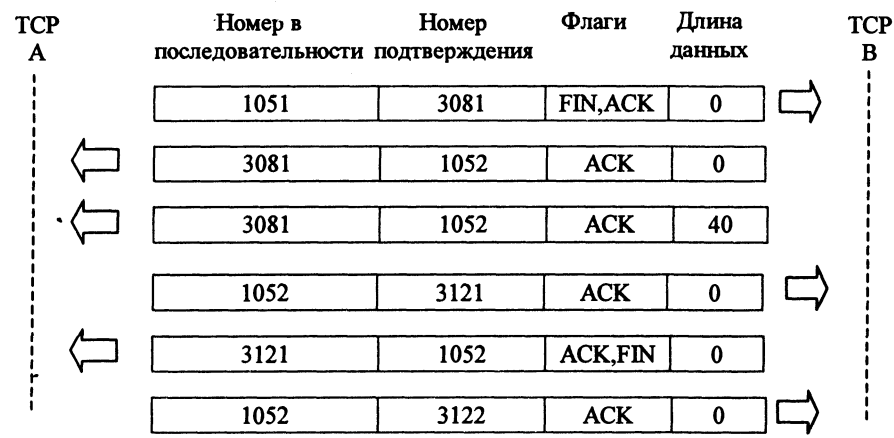


Рис. 5.52. Закрытие соединения

Таймеры и их назначение. *Таймер повторной передачи.* Данный таймер устанавливается значением RTO (Retransmission Time Out – интервал до повторной передачи) в момент посылки TCP-пакета адресату. Если таймер будет сброшен в ноль до момента получения подтверждения пакета, то этот пакет должен быть послан вновь.

Ясно, что величина RTO не может быть фиксированной, так как TCP-пакеты до разных адресатов следуют по различным маршрутам через сети, скорость передачи данных в которых может различаться более чем в тысячи раз. Для вычисления «оптимального» значения RTO в каждом логическом соединении используется специальная процедура, специфицированная в RFC-793. Согласно этой процедуре, для каждого TCP-пакета измеряется величина RTT (Round Trip Time – интервал времени от момента посылки TCP-пакета до момента получения подтверждения на него). По измеренным RTT определяют величину SRTT (Smoothed RTT – сглаженный RTT):

$$SRTT = k \times SRTT + (1 - k) \times RT. \qquad (5.1)$$

Здесь k – сглаживающий коэффициент (например, 0,9).

Формула (5.1) учитывает фильтрацию нетипичных (пиковых) значений измеренной величины RTT.

«Оптимальное» значение RTO вычисляют по формуле:

$$RTO = \min(U, \max(L, p \times SRTT)), \qquad (5.2)$$

где: U – ограничение сверху на значение RTO (например, 30 с); L – ограничение снизу на значение RTO (например, 1 с); p – коэффициент «запаса» (например, 2).

Если после повторной посылки TCP-пакета, вновь не будет получено его подтверждение за интервал времени RTO, то попытки послать TCP-пакеты будут повторены (до 12 раз), но каждый раз с экспоненциально возрастающим значением RTO. Только после неудачи всей серии повторных посылок связь между партнерами будет считаться аварийно закрытой.

Таймер возобновления передачи. В ходе взаимодействия двух TCP-модулей (A и B) вполне возможна следующая ситуация:

- TCP-модуль B уведомляет TCP-модуль A о невозможности приема от него данных, определяя размер окна равным 0;
- TCP-модуль A, имея данные для передачи, переходит в состояние ожидания от TCP-модуля B пакета с ненулевым размером окна;
- TCP-модуль B, у которого освободилось некоторое пространство в буферах, посылает модулю A TCP-пакет с ненулевым размером окна;
- адресованный модулю A пакет «потерян» по какой-либо причине и оба TCP-модуля переходят в состояние бесконечного ожидания.

Для выхода из такого тупикового состояния и служит таймер возобновления передачи (persistence timer – «настойчивый» таймер). Он устанавливается в момент получения *TCP*-пакета с нулевым значением поля «Размер окна» в его

заголовке (типичное начальное значение для этого таймера – 5 с). Если до момента обнуления таймера не будет получено разрешение на возобновление передачи данных, то ожидающий разрешения TCP-модуль отправляет партнеру пакет, содержащий всего лишь 1 байт данных. По реакции партнера, возвращающего пакет с нулевым/ненулевым значением размера окна, TCP-модуль продолжает ожидание или возобновляет посылку данных.

Таймер закрытия связи. Протокол TCP предусматривает следующий простой прием предотвращения появления в сети TCP-пакетов, не имеющих адресатов: после закрытия логического соединения между партнерами номера портов, использовавшихся в этом соединении, остаются еще некоторый интервал времени действительными, что дает возможность долго блуждавшим по сети TCP-пакетам добраться до места назначения (где они будут просто проигнорированы). Значение этого интервала равно удвоенному времени жизни IP-пакета (обычно, 2 ×15 = 30 с).

Таймеры поддержки соединения. Для проверки наличия логического соединения между TCP-модулями используют следующий механизм. Каждый TCP-модуль, участвующий в логическом соединении, через фиксированный промежуток времени (keep-alive timer), равный обычно 45 с, периодически отправляет партнеру пустые (не содержащие данных) TCP-пакеты и ждет их подтверждения. Каждое полученное подтверждение говорит о сохранении соединения. Если же в течении определенного интервала времени (idle timer), равного обычно 360 с, не будет получено ни одного подтверждения, то логическое соединение считается оборванным.

Очевидно, что данный механизм имеет смысл включать в работу только тогда, когда партнеры по TCP-взаимодействию приостановили по какой-либо причине обмен данных на достаточно длительный срок (более 45 с).

Протокол UDP

Протокол дейтаграмм пользователя UDP (User Datagram Protocol) является протоколом транспортного уровня и базируется на возможностях, предоставляемых межсетевым протоколом IP. Основная задача TCP – обеспечение «быстрой» передачи данных в сети. Его транспортный адрес в заголовке IP-сегмента равен 17. Описание протокола UDP дано в рекомендации RFC-768.

Назначение и основные характеристики протокола. Протокол транспортного уровня UDP играет роль интерфейса для прикладных программ к средствам протокола межсетевого уровня IP. Данные, отправляемые прикладным процессом через модуль UDP, достигают места назначения как единое целое. Например, если процесс отправитель производит 6 записей в UDP-порт, то процесс-получатель должен будет сделать 6 чтений. Размер каждого записанного сообщения будет совпадать с размером каждого прочитанного. Протокол UDP сохраняет границы сообщений, определяемые прикладным процессом. Он никогда не объединяет несколько сообщений в одно и не делит одно сообщение на части.

Присвоение номера порта можно осуществить любым из рассмотренных выше способов, в том числе и локальным, который состоит в том, что разработчик некоторого приложения просто связывает с портом любой доступный, произвольно выбранный числовой идентификатор, обращая внимание на то, чтобы он не входил в число зарезервированных номеров портов. В дальнейшем все удаленные запросы к данному приложению от других приложений должны быть адресованы с указанием назначенного ему номера порта.

Основные характеристики протокола:

- реализует взаимодействие без установления логического (виртуального) соединения;
- организует дейтаграммную передачу данных;
- использует 16-битовые номера портов для идентификации партнеров по взаимодействию на транспортном уровне;
- не гарантирует надежной передачи данных (возможна как потеря UDP-пакетов, так и их дублирование);
- не имеет средств уведомления источника UDP-пакета о доставке пакета адресату;
- не обеспечивает правильный порядок доставки UDP-пакетов от источника к приемнику;
- может гарантировать целостность данных в UDP-пакете за счет использования контрольной суммы;
- очень прост (особенно, по сравнению с протоколом TCP).

Формат UDP-дейтаграммы и назначение полей. Формат заголовка для дейтаграмм пользователя приведён на рис. 5.53. Если задействован *порт отправителя*, то он указывает порт процесса, посылающего дейтаграмму. Можно принять, что это тот порт, на который при отсутствии какой-либо иной информации следует адресовать ответную дейтаграмму. Если данное поле не задействовано, то в него следует записать нули. *Порт получателя* имеет смысл только в контексте конкретного Internet-адреса получателя.

Длина – длина в октетах данной UDP-дейтаграммы, включающая как заголовок, так и данные (это означает, что минимальное значение поля длины равно восьми).

Контрольная сумма – 16-битное дополнение до единицы суммы дополнений полей UDP-заголовка, поля данных, нескольких полей из заголовка в протоколе IP (IP-адрес отправителя, IP-адрес получателя, поле протокола) и поля длины UDP-дейтаграммы. Это означает, что UDP должен взаимодействовать с IP для нахождения нужных адресов перед посылкой дейтаграммы.

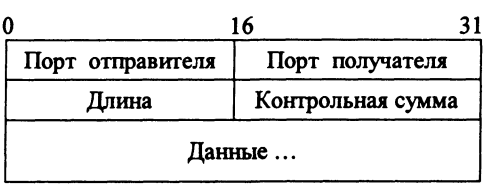


Рис. 5.53. Формат заголовка UDP-дейтаграммы

Целью использования IP-адресов в контрольном суммировании является проверка того, что UDP-дейтаграмма достигла своего настоящего

места назначения, так как UDP-заголовок определяет только номер протокольного порта. Таким образом, чтобы проверить место назначения, UDP на компьютере-источнике вычисляет контрольную сумму, которая учитывает IP-адрес назначения, а так же саму UDP-дейтаграмму. При получении дейтаграммы в месте назначения программы UDP проверяют контрольную сумму, используя IP-адрес назначения, полученный из заголовка IP-дейтаграммы, которая содержала UDP-сообщение. Если контрольные суммы одинаковы, дейтаграмма действительно достигла нужного хост-компьютера и нужного порта в нем.

Если контрольная сумма равна нулю, то это означает, что отправитель дейтаграммы ее не подсчитывал, и, следовательно, ее нужно игнорировать. Если два модуля UDP взаимодействуют только через одну сеть Ethernet, то от контрольного суммирования можно отказаться, так как средства Ethernet обеспечивают достаточную степень надежности обнаружения ошибок передачи. Это снижает накладные расходы, связанные с работой UDP. Однако рекомендуется всегда выполнять контрольное суммирование, так как возможно в какой-то момент изменения в таблице маршрутов приведут к тому, что дейтаграммы будут посылаться через менее надежную среду.

Если контрольная сумма правильная (или равна нулю), то проверяется порт назначения, указанный в заголовке дейтаграммы. Если к этому порту подключен прикладной процесс, то прикладное сообщение, содержащееся в дейтаграмме, встает в очередь для прочтения. В остальных случаях дейтаграмма отбрасывается. Если дейтаграммы поступают быстрее, чем их успевает обрабатывать прикладной процесс, то при переполнении очереди сообщений поступающие дейтаграммы отбрасываются модулем UDP.

UDP-сообщения, включающие UDP-заголовок и данные, инкапсулируются в IP-дейтаграммах при передаче по сети.

Мультиплексирование и демультиплексирование прикладных протоколов с помощью протокола UDP. Протокол UDP ведет для каждого порта две очереди: очередь пакетов, поступающих в данный порт из сети, и очередь пакетов, отправляемых данным портом в сеть. Процедура обслуживания протоколом UDP-запросов, поступающих от нескольких различных прикладных сервисов, называется *мультиплексированием*. Распределение протоколом UDP поступающих от сетевого уровня пакетов между набором высокоуровневых сервисов, идентифицированных номерами портов, называется *демультиплексированием*.

Концептуально все процессы мультиплексирования и демультиплексирования между UDP и прикладными программами осуществляются с помощью механизма портов. На практике каждая прикладная программа должна договориться с операционой системой о получении протокольного порта и связанного с ним номера перед посылкой UDP-дейтаграммы. Когда порт выделен, прикладная программа посылает любую дейтаграмму через порт, номер которого указан в поле «Порт отправителя» UDP-заголовка. В ходе обработки входных данных

UDP принимает приходящие от IP дейтаграммы и демультиплексирует их по портам назначения. UDP использует номер порта получателя UDP для выбора соответствующего получателя для пришедшей дейтаграммы.

Порт UDP легче всего представить в виде очереди. В большинстве реализаций, когда прикладная программа договаривается с операционой системой об использовании данного порта, операционная система создает внутреннюю очередь, которая хранит приходящие сообщения. Часто приложение может указать или изменить размеры очереди. Когда UDP получает дейтаграмму, он проверяет, нет ли порта назначения с таким номером среди используемых портов. Если нет, он посылает ICMP-сообщение об ошибке «порт недоступен» и уничтожает дейтаграмму. Если есть, UDP добавляет новую дейтаграмму в очередь порта, где прикладная программа может ее получить. Конечно, если очередь порта уже переполнена, то тогда UDP уничтожает новую дейтаграмму.

К услугам протокола UDP может обратиться любое приложение, однако многие из них предпочитают иметь дело с более сложным протоколом транспортного уровня – TCP. Дело в том, что протокол UDP выступает простым посредником между сетевым уровнем и прикладными сервисами, и, в отличие от TCP, не берет на себя никаких функций по обеспечению надежности передачи. Протокол UDP, как уже отмечалось, является дейтаграммным протоколом, т. е. он не устанавливает логического соединения, не нумерует и не упорядочивает пакеты данных.

С другой стороны, функциональная простота протокола UDP обусловливает простоту его алгоритма, 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, каким является протокол UDP. Протокол UDP можно использовать и в том случае, когда хорошее качество каналов связи обеспечивает достаточный уровень надежности и без применения дополнительных приемов типа установления логического соединения и квитирования передаваемых пакетов.

5.7. Протоколы I уровня стека TCP/IP

Протокол FTP

Протокол пересылки файлов FTP (File Transfer Protocol) реализует удаленный доступ к файлу. Для того чтобы обеспечить надежную передачу, FTP использует в качестве транспорта протокол с установлением соединений – TCP. Кроме пересылки файлов протокол FTP предлагает и другие услуги. Так, пользователю предоставляется возможность интерактивной работы с удаленной машиной, например, он может распечатать содержимое ее каталогов. Наконец, FTP выполняет аутентификацию пользователей. Прежде чем получить доступ к файлу, в соответствии с протоколом пользователи должны сообщить свое

имя и пароль. Для доступа к публичным каталогам FTP-архивов Internet парольная аутентификация не нужна, и ее обходят за счет использования для такого доступа предопределенного имени пользователя Anonymous.

В стеке TCP/IP протокол FTP предлагает наиболее широкий набор услуг для работы с файлами, однако он является и самым сложным для программирования. Приложения, которым не нужны все возможности FTP, могут использовать другой, более экономичный протокол – простейший протокол пересылки файлов TFTP (Trivial File Transfer Protocol). Этот протокол реализует только передачу файлов, причем здесь транспортом выступает более простой, чем TCP, протокол без установления соединения – UDP.

Протокол telnet

Протокол telnet обеспечивает передачу потока байтов между процессами, а также между процессом и терминалом. Наиболее часто этот протокол применяют для эмуляции терминала удаленного компьютера. При использовании сервиса telnet пользователь фактически управляет удаленным компьютером так же, как и локальный пользователь, поэтому такой вид доступа требует хорошей защиты. Поэтому серверы telnet всегда используют как минимум аутентификацию по паролю, а иногда и более мощные средства защиты, например, систему Kerberos.

Протокол SNMP

Протокол SNMP (Simple Network Management Protocol) используют для организации сетевого управления. Изначально протокол SNMP был разработан для удаленного контроля и управления маршрутизаторами Internet, которые традиционно часто называют также шлюзами. С ростом популярности протокол SNMP стали применять и для управления коммуникационным оборудованием – концентраторами, мостами, сетевыми адаптерами и т. д. Управление в протоколе SNMP решает две задачи.

Первая задача связана с передачей информации. Протоколы передачи управляющей информации определяют процедуру взаимодействия SNMP-агента, работающего в управляемом оборудовании, и SNMP-монитора, работающего на компьютере администратора, который часто называют также консолью управления. Протоколы передачи определяют форматы сообщений, которыми обмениваются агенты и монитор.

Вторая задача связана с контролируемыми переменными, характеризующими состояние управляемого устройства. Стандарты регламентируют, какие данные должны сохраняться и накапливаться в устройствах, имена этих данных и синтаксис этих имен. В стандарте SNMP определена спецификация информационной базы данных управления сетью. Эта спецификация, известная как база данных MIB (Management Information Base), определяет те элементы данных, которые управляемое устройство должно сохранять, и допустимые операции над ними.

6. СИСТЕМЫ ЭЛЕКТРОННОЙ ПОЧТЫ И ПОЧТОВЫХ КАТАЛОГОВ

В этой главе рассмотрены основы построения систем электронной почты и почтовых каталогов; возможности и общие сведения о системах, построенных на базе рекомендации X.400 и протокола SMTP, на базе частных стандартов MS Mail, cc:Mail; гибридные системы электронной почты. Приведены базовые сведения о рекомендации X.500 и почтовых каталогах частных фирм, описана взаимосвязь между ними и облегченный протокол доступа к каталогам LDAP.

6.1. Системы на базе стандарта X.400

Стандарт X.400 представляет собой набор рекомендаций по построению системы передачи электронных сообщений, не зависящей от используемых на сервере и клиенте операционных систем и аппаратных средств. Рекомендации X.400 являются результатом деятельности международного комитета по средствам телекоммуникаций (CCITT во французской транскрипции или ITU – в английской), созданного при Организации Объединенных Наций. Они охватывают все аспекты построения среды управления сообщениями: терминологию, компоненты и схемы их взаимодействия, протоколы управления и передачи, форматы сообщений и правила их преобразования. В рекомендациях X.400 наиболее полно отражен накопленный в индустрии компьютеров и телекоммуникаций опыт создания и применения информационных систем.

Рекомендации X.400 опираются на семиуровневую модель и семейство протоколов OSI/ISO. Это обеспечивает системам, построенным на основе такой модели, высокую степень независимости от среды передачи данных. Поскольку рекомендации X.400 определяют набор спецификаций для самого верхнего Прикладного уровня, отвечающие этим рекомендациям приложения должны свободно взаимодействовать друг с другом, вне зависимости от применяемых операционных систем, аппаратуры и сетевых протоколов.

Первые четыре уровня (от физического до транспортного) отвечают за организацию среды передачи данных. Реализуются частично на аппаратном уровне, частично как сервисы ядра операционной системы. Верхние три уровня предназначены для использования операционной системой и исполняющимися поверх нее прикладными программами.

Для разделения входящего потока данных между приложениями на каждом из уровней, транспортном, сеансовом и представительном, использован механизм так называемых точек доступа (SAP – Service Access Point). Каждая точка доступа имеет уникальный идентификатор, который представляет собой либо символьную строку, либо последовательность шестнадцатеричных цифр. Длина идентификатора транспортного уровня составляет 32 символа (64 цифры), уровня сеансов – 16 символов (32 цифры) и представительного уровня – 8 символов (16 цифр). Чтобы два приложения в сети могли взаимодействовать, каждое из них должно знать набор идентификаторов другого.

Основные компоненты

Рассмотрим подробнее определение компонентов системы передачи электронных сообщений в терминах X.400. В классической постановке *среда управления сообщениями* (MHE – Messaging Handling Environment) представляет собой объединение *систем управления сообщениями* (MHS – Messaging Handling Systems), которые могут быть произвольным образом связаны между собой посредством шлюзов и/или публичных информационных сетей. Каждая из систем управления сообщениями в свою очередь состоит из следующих компонентов:

- *пользовательский агент* (UA – User Agent), подсистема, выступающая в роли клиента в процессе обмена почтовыми сообщениями. Клиент же, в свою очередь, может быть как реальным пользователем, так и процессом, использующим сервисы электронной почты;
- *агент передачи сообщений* (MTA – Message Transfer Agent), подсистема, в обязанности которой входит обмен сообщениями с пользовательскими агентами и/или внешними и локальными агентами передачи сообщений. Каждый агент передачи сообщений может иметь имя и пароль доступа;
- *система передачи сообщений* (MTS – Message Transfer System) выполняет функции приема, доставки и промежуточного хранения сообщений; состоит из одного или нескольких MTA;
- *хранилище сообщений* (MS – Message Store), подсистема, в функции которой входит посылка, прием и хранение сообщений от пользовательских агентов и агентов передачи сообщений, в составе MHS может быть более одного хранилища.

Из всего многообразия описанных в рекомендациях X.400 способов взаимодействия между UA, MTA и MS рассмотрим лишь те, которые любая из ныне известных почтовых систем использует для отправки и приема сообщений:

отправка сообщения пользовательским агентом через хранилище. Пользователь, используя свой агент (UA), помещает сообщение, предназначенное для доставки другому пользователю, непосредственно в хранилище сообщений. Оттуда оно забирается локальным или удаленным MTA и передается дальше;

отправка сообщения пользовательским агентом через MTA. Сообщение передается напрямую от UA к MTA, который далее осуществляет доставку своими средствами;

получение сообщения агентом пользователя из хранилища. MTA осуществляет доставку сообщения в хранилище (почтовый ящик пользователя) для дальнейшей обработки UA;

получение сообщения агентом пользователя от MTA. Пользовательский агент не имеет непосредственного доступа к хранилищу и для получения сообщений он должен обратиться к агенту передачи.

Не менее важными компонентами спецификации X.400 являются следующие компоненты:

• *списки рассылки* (DL – Distribution Lists), содержащие ноль или более членов, каждый из которых может быть либо пользователем системы управления сообщениями, либо другим списком рассылки. Будучи отправленным на адрес списка рассылки, сообщение будет доставлено всем его членам, включая вложенные списки пользователей;

• *устройство доступа* (AU – Access Unit) – устройство, больше известное как шлюз (Gateway), обеспечивающее сопряжение с внешней средой передачи данных, например с телекс- или телетайп-сетями;

• *каталог* (Directory), его основное назначение заключается в хранении информации об объектах, входящих в состав системы управления сообщениями. Реализация этой части в системе X.400 является необязательной.

Дополнительно в состав MHS могут входить следующие компоненты, которые не являются специфическими для X.400 и определены. В отдельной спецификации X.500 определены дополнительные компоненты, которые не являются обязательными для рекомендации X.400, но могут входить в состав MHS. Это следующие компоненты:

• *пользовательский агент доступа к каталогу* (DUA – Directory User Agent), подсистема, выступающая в роли клиента при доступе к каталогу;

• *системный агент доступа к каталогу* (DSA – Directory System Agent), подсистема, являющаяся частью каталога и предоставляющая доступ к хранящейся в нем информации локальным и внешним DUA и DSA.

Базовым понятием в рекомендациях X.400 является почтовое сообщение и его составляющие. Для описания формата сообщения в рекомендациях X.400 была принята привычная парадигма *конверта* (envelope) и *содержимого* (content) традиционных почтовых систем. Как и положено, конверт содержит исчерпывающую информацию о том, куда и кому должно быть доставлено письмо, обратный адрес отправителя и пометку о срочности доставки. При этом системе нет необходимости знать, что бы то ни было о содержимом письма.

На основе информации, указанной на конверте, среда доставки выполняет необходимую маршрутизацию и передачу с возможным промежуточным хранением (store and forward). Роль перевалочных пунктов и средств транспортировки выполняют MTA. Конверт может иметь специальную пометку о необходимости установки на нем электронного «штампа» (trace information) каждым MTA, через который проходит сообщение на пути к адресату. Это, в частности, позволяет системе автоматически отслеживать возникновение почтовых петель. Формат конверта X.400 определяет спецификация P1.Формат содержимого определяется его функциональной нагрузкой. Поскольку основная функция MTS состоит в передаче сообщений между людьми (персонами), для этого существует специальный тип *содержимого*, называемый интерперсональным сообщением (IPM – Inter Personal Message). Интерперсональное сообщение представляет собой составной объект – IPM состоит из *заголовка* (header) и *тела* (body). Заголовок обычно включает в себя копию информации, указанной на конверте, и дополнительных полей, определяющих расширенные свойства сообщения. Тело, в свою очередь, может быть составным и включать различные типы информации, такие, как плоский текст, графика, документы различных форматов, вложенные сообщения и т. д. Отдельные части сообщения именуются body parts. В настоящее время используют два формата IPM, различающиеся набором поддерживаемых типов данных и правил кодирования текста, содержащего символы национальных алфавитов: P2, используемый в системах X.400 1984 г., и P22, используемый в системах X.400 1988 г. Системы 1988 г. могут работать как с представлением данных в формате P2, так и P22.

Еще один тип содержимого сообщений X.400 – интерперсональная нотификация (IPN – Inter Personal Notification). Нотификацию используют для автоматического уведомления отправителя о факте доставки и/или прочтения, посланного им сообщения. IPN представляет собой плоский текст произвольного содержания в формате US-ASCII. Прочие типы содержимого сообщений несут служебную нагрузку и используются исключительно для взаимодействия систем между собой.

Несмотря на мощную теоретическую базу и практически безупречный архитектурный дизайн, семейство протоколов X.400 не получило широкого распространения за пределами государственных и банковских учреждений. «Ахиллесовой пятой» этого стандарта явились чрезмерная сложность реализации и значительная стоимость внедрения и эксплуатации систем на его основе. Отсутствие свободного доступа к стандартам и проблемы несовместимости MTA различных версий также отрицательно сказались на темпах внедрения X.400 в качестве глобальной среды передачи данных.

Адресация в X.400

В системах на базе рекомендаций X.400 используется одна из самых мощных схем адресации, известная как автор/получатель (O/R – Originator/Recipient). Структура адреса и терминология, применяемая при определении адресов, опи-

рается на предположение (не лишенное оснований), что глобальная телекоммуникационная сеть управляется и поддерживается официально зарегистрированными в CCITT/ITU коммерческими компаниями, предоставляющими свои услуги прочим организациям. В терминах рекомендаций X.400 телекоммуникационные компании называются администрацией (Administration). Управляющим доменом (MD – Management Domain) называется объединение по крайней мере одного MTA и произвольного (в том числе нулевого) числа пользовательских агентов (UA), информационных хранилищ (MS) и/или шлюзов (AU), принадлежащих и управляемых одной компанией. Управляющий домен, поддерживаемый администрацией, называется административным управляющим доменом (ADMD – Administration Management Domain). Остальные домены, обслуживаемые неадминистрациями, называются частными управляющими доменами (PRMD – Private Management Domain). В обязанности ADMD входит контроль за уникальностью имен PRMD, пользующихся его услугами, обеспечение корректной работы телекоммуникационного оборудования, начисление платы за услуги и взаимные расчеты с другими ADMD. PRMD назначает имена внутри собственного управляющего домена. Согласно рекомендациям CCITT, частные управляющие домены должны направлять весь нелокальный трафик только через свой административный домен, прямая же передача данных между PRMD не рекомендуется. На территории каждого государства может существовать несколько ADMD, однако в целях обеспечения «максимальной» совместимости с национальной политикой сфера деятельности ADMD не распространяется за пределы государственных границ. По этой же причине существование международных PRMD неявно запрещено.

Пользовательский адрес X.400 представляет собой набор атрибутов. Для разделения атрибутов используют либо прямой слеш, либо двоеточие. Каждый атрибут записывается в виде КЛЮЧЕВОЕ_СЛОВО=ЗНАЧЕНИЕ, для ключевых слов можно использовать аббревиатуры и метки. Часть атрибутов, не оказывающих влияния на уникальность адреса, можно опустить. Сведения об адресате могут иметь произвольный порядок следования. Существует четыре типа адресов X.400:

- *мнемонический* (Mnemonic) – для представления обычных пользователей и списков рассылки (этот тип адресов используется наиболее часто);
- *цифровой* (Numeric), служит для представления пользователей, использующих только цифровые клавиатуры для регистрации и посылки сообщений;
- терминальный (Terminal) – для представления пользователей, использующих терминалы, подключенные к сетям передачи данных;
- *почтовый* (Postal) – для представления пользователей, не использующих электронных устройств.

Маршрутизация в X.400

В силу архитектурных особенностей X.400 для гарантированного установления соединения между двумя MTA необходима ручная настройка значитель-

ного числа параметров, таких, как идентификаторы SAP, имена и пароли MTA и т. п. Поэтому динамическая маршрутизация в системах X.400 невозможна. Однако в случае использования каталога организации, информация о маршрутах может быть добавлена в таблицы автоматически.

6.2. Системы на базе протокола SMTP

SMTP (Simple Message Transfer Protocol), или в дословном переводе простой протокол передачи сообщений, был разработан в среде UNIX и предназначался исключительно для общения между собой почтовых серверов. В модели OSI протокол SMTP хотя и находится на уровне приложений, способен общаться только с TCP/IP, расположенном на четвертом транспортном уровне.

Общие сведения и характеристика протокола SMTP

В связи с бурным ростом Internet протокол SMTP на сегодняшний день получил очень широкое распространение как протокол передачи сообщений. Практически все производители пакетов электронной почты либо поддерживают протокол SMTP как базовый, либо используют его на уровне шлюзов. В большой степени такая популярность объясняется сравнительной простотой реализации и широкими возможностями расширяемости без ущерба для обратной совместимости с существующими версиями почтовых систем. Немаловажным фактором является также широкая доступность спецификаций и отсутствие необходимости отчислять средства за их использование.

SMTP-системы в последнее время активно развиваются в следующих направлениях:

- расширение протокола общения сервер-сервер (собственно SMTP);
- создание и улучшение протокола общения клиент-сервер (POP3, IMAP4);
- внедрение и расширение нового формата сообщений (MIME).

Начальная версия протокола SMTP поддерживала ограниченный набор команд и сервисов для приема и передачи сообщений. В последнее время был разработан его расширенный вариант (Extended или ESMTP), обеспечивающий стандартную возможность дальнейшего расширения и поддержку таких функций, как подтверждение доставки (DNR – Delivery Notification Request), согласование максимального допустимого размера сообщений, передаваемых между серверами, и принудительная инициация передачи накопленной почты (dequeue). Одним из недостатков SMTP на данный момент является отсутствие возможности аутентификации входящих соединений, шифрования диалога и потока передачи данных между серверами.

Отсутствие средств аутентификации входящих соединений не позволяет использовать SMTP для обслуживания клиентского доступа. Классическая почтовая SMTP-система требует наличия файлового доступа клиента к своему почтовому ящику для получения и работы с сообщениями. Для реализации работы в режиме клиент-сервер был создан протокол обслуживания почтового

офиса (POP – Post Office Protocol). Самой удачной оказалась версия POP3, широко используемая в современных SMTP-системах. Наиболее совершенные реализации поддерживают аутентификацию с шифрованием имени и пароля и шифрование трафика по протоколу Secure Socket Layer (SSL). Однако при использовании протокола POP3 невозможен просмотр характеристик сообщения без предварительной загрузки его на станцию клиента. Для просмотра и манипуляции свойствами почтового сообщения непосредственно на сервере, а также преодоления ряда других функциональных ограничений был разработан протокол IMAP4. Следует заметить, что как для случая использования классического клиента (команда mail), так и для случая применения POP3 или IMAP4 отправка подготовленных клиентом сообщений требует наличия сервера SMTP.

Изначально SMTP-системы предназначались для передачи информации исключительно в текстовом виде и не были ориентированы на передачу символов национальных алфавитов, т. е. использовали 7-битный набор символов. Для передачи двоичных файлов был разработан стандарт UUENCODE, позволяющий внедрять предварительно преобразованные из бинарного в текстовый вид произвольные данные непосредственно в текст сообщения. Даже в этом случае никакой информации о типе передаваемых данных и породившем их приложении принимающая сторона не имела. По мере расширения сети Internet, усложнения программного обеспечения и активного внедрения мультимедиа назрела необходимость создания универсального формата типизации и представления двоичных данных и текста, содержащего национальные символы. Таким универсальным форматом стали многофункциональные расширения почты Internet (MIME – Multipurpose Internet Mail Extensions). Формат MIME оказался удобным, поскольку в него были заложены возможности неограниченного расширения как поддерживаемых типов данных, так и национальных кодировок.

Сообщение SMTP, подобно сообщению X.400, использует понятия конверта и содержимого, которое, в свою очередь, имеет *заголовок* и *тело*. Функциональное назначение их полностью идентично. Состав полей в заголовке определяется форматом тела сообщения (UUENCODE или MIME). Ни одно поле не является обязательным, но, как правило, указываются поля: *кому* (To:), *от кого* (From:) и *тема* (Subject:). В случае использования формата MIME в заголовке обязательно должна присутствовать строка «MIME-Version: 1.0». Полный перечень возможных полей в заголовке сообщения SMTP содержится в рекомендации RFC 2076.

Отличительной особенностью SMTP-систем является то, что в них, как правило, обеспечена фактическая независимость процесса передачи от формата *содержимого*. За интерпретацию содержимого должна отвечать только клиентская программа (mail reader). Однако платой за совместимость на уровне MTA в данном случае является неэффективность передачи любых нетек-

стовых данных или сообщений, использующих символы национальных алфавитов, вследствие предварительной трансляции информации в текстовое представление. В зависимости от используемого алгоритма преобразования размер фактически передаваемых данных может возрасти до 100 %.

Немаловажной проблемой при передаче данных через SMTP-системы является обеспечение конфиденциальности. Поскольку сообщения передаются в текстовом виде, они могут быть легко перехвачены и произвольным образом изменены. Для решения проблем защиты информации был создан протокол на шифрование тела сообщения, так называемые засекреченные многофункциональные расширения почты (Secure MIME или S/MIME). Однако этот протокол не защищает от перехвата заголовки сообщений.

Адресация в SMTP

В системах на базе SMTP используется интуитивно понятная, простая и одновременно очень мощная иерархическая схема адресации, аналогичная той, что принята в службе имен Internet (DNS – Domain Name Services). Данная схема обеспечивает уникальность адреса практически неограниченному числу пользователей. Почтовый адрес SMTP записывают в следующем виде: mailbox@domain, где mailbox – символическое имя почтового ящика пользователя (до 63 символов); domain – уникальное имя (почтовый домен) системы, в которой зарегистрирован упомянутый пользователь (до 255 символов).

Сочетание имени и домена образует уникальный идентификатор пользователя. Почтовый домен хранит полную информацию о положении системы в иерархии почтового пространства организации. Каждый следующий уровень иерархии отделен от предыдущего точкой. Разбор имени домена происходит справа налево. Самый верхний уровень (top level), называемый корневым доменом (root domain), соответствует либо типу организации (com – для коммерческой, gov – для государственной, org – для общественной и т. п.), либо географическому региону (стране) (ru – для России, fr – для Франции и т. д.). Следующими в иерархии идут домены первого уровня (first level), как правило, представляющие имя организации. Регистрацией имен доменов первого уровня занимается международный центр Internet (InterNIC – Internet Network Information Center). За назначение имен доменов более низкого уровня чаще всего отвечают сами компании. Поскольку организациям не запрещено регистрировать для собственных нужд несколько параллельных доменов, например CIT.MSK.RU и CIT.COM, пользователь может иметь более одного SMTP-адреса. В приведенном примере CIT.MSK.RU является субдоменом MSK.RU, который, в свою очередь, выступает субдоменом RU. Компания CIT имеет два зарегистрированных имени, и следовательно каждый пользователь может иметь два почтовых адреса. Кроме того, современные SMTP-системы зачастую позволяют назначать псевдонимы для самого почтового ящика (например JohnDoe@CIT.MSK.RU и JonnyD@CIT.MSK.RU).

Маршрутизация в SMTP

Для того чтобы SMPT-сервер доставил почту на имя адресата JohnDoe@CIT.MSK.RU, ему предварительно нужно узнать IP-адрес машины, обслуживающей почтовый домен CIT.MSK.RU, обратившись с соответствующим запросом к серверу DNS. В службе имен DNS предусмотрен специальный тип ресурсной записи для обслуживания такого рода запросов – MX или Mail Exchanger, т. е. сервер, выполняющий обмен почтовыми сообщениями от имени домена. Упомянутая запись имеет следующий формат:

domain MX [cost] hostname

где domain – это имя почтового домена, к которому принадлежит адресат; hostname – символическое имя компьютера, располагающего знаниями о том, как осуществлять дальнейшую доставку (для получения IP-адреса компьютера с именем hostname выполняется поиск адресной ресурсной записи в DNS); cost – относительная стоимость доставки через этот компьютер.

При наличии нескольких MX-записей для одного и того же домена сначала будет выполнена попытка установить соединение с тем компьютером, у которого стоимость доставки ниже. Если этот компьютер окажется недоступен или перегружен, будут использованы компьютеры с большими значениями стоимости.

Таким образом, чтобы доставить сообщение на имя адресата John Doe@CIT.MSK.RU, сначала будет выполнен запрос к серверу DNS на получение списка ресурсных записей с типом MX. Если список не пуст, по имени компьютера с наименьшим значением стоимости доставки будет получен его адрес (опять же через DNS), после чего будет установлено соединение и отправлена почта. Если для домена CIT.MSK.RU нет MX-записи, домен будет трактоваться как имя компьютера. Будет выполнена попытка получить IP-адрес компьютера и доставить сообщение напрямую.

Несмотря на то, что DNS-имена назначаются заранее и являются статической информацией, маршрутизацию сообщений SMTP в Internet можно рассматривать как динамическую, так как следующая точка маршрута (теоретически) должна определяться заново для каждого сообщения.

В сетях, не имеющих прямого выхода в Internet и не использующих возможности MX-записей DNS, часто используют статическую схему маршрутизации. Практически все существующие SMTP-системы позволяют применять языки сценариев для описания статических таблиц.

Согласно терминологии, принятой в Internet, SMTP-сервер может выступать в одной (или нескольких) из следующих ролей:

- mail exchanger – компьютер, непосредственно подключенный к Internet и выполняющий доставку сообщений напрямую адресатам внутри организации, к которой он принадлежит. В организации может быть несколько таких компьютеров с различными или одинаковыми значениями показателя стоимости доставки;

• relay – компьютер, выполняющий прием почтового трафика от лица других доменов, не имеющих непосредственного и/или постоянного подключения к Internet и, как правило, не принадлежащий к организациям, чьи домены он обслуживает. Как косвенно следует из вышесказанного, для каждого отдельного домена может быть определено не более одного relay-сервера;

• smart host – компьютер, способный осуществлять пересылку сообщений на основе собственной статической таблицы маршрутизации. Одной из функций smart host является переписывание на конверте адреса получателя и/или отправителя перед осуществлением дальнейшей передачи сообщения.

Большинство современных реализаций SMTP-серверов позволяет сочетать все перечисленные функции на одном компьютере.

6.3. Системы на основе частных стандартов (MS Mail, cc:Mail)

Особенности построения и основные характеристики

Параллельно с развитием персональных компьютеров и сетей на их основе развивались системы электронной почты, использующие файловый метод доступа к информационным хранилищам, собственные форматы сообщений и протоколы взаимодействия агентов передачи сообщений. Классическим примером таких систем могут служить Microsoft Mail for PC Networks и Lotus cc:Mail. До начала массового распространения SMPT- и X.400-систем электронные почты на основе патентованных стандартов были весьма популярны и широко использовались. Это объясняется тем, что, не имея такой сложности реализации и внедрения, как почты X.400, они обладали гораздо большей функциональностью и были гораздо удобнее в работе, чем SMTP-системы. Например, каждая из частных систем предоставляла своим пользователям такие сервисы, как поддержка вложенных списков рассылки, подтверждений о прочтении сообщения, множественных хранилищ (общих и личных папок) и средств группового планирования. К тому же для их работы не требовалось наличия на рабочих местах протокола TCP/IP и дорогостоящих UNIX-серверов. Кроме того, они хорошо работали в любых локальных сетях. Наличие шлюзов в другие почтовые системы обеспечивало и продолжает обеспечивать им достаточно гладкую интеграцию в единое почтовое пространство многих компаний. До настоящего времени эти системы успешно работают в организациях со сравнительно небольшим числом сотрудников (до 300). Следует упомянуть, что результатом развития именно систем на основе частных стандартов стало появление повсеместно используемых наборов интерфейсов прикладных программ, таких, как MAPI (Messaging API) и VIM (Vendor Independent Messaging). Их поддержка реализована на сегодня практически во всех клиентских программах работы с электронной почтой.

Однако у систем рассматриваемого типа есть ряд существенных недостатков. Все они используют для организации хранилища сообщений парадигму *почтового отделения* (PO – Post Office). Почтовое отделение представляет собой набор файлов и каталогов определенной структуры, располагаемых на разделяемом ресурсе файлового сервера. Для такой схемы размещения необ-

ходимо наличие прав на запись и удаление для каждого пользователя на соответствующем разделяемом ресурсе, что делает их чрезвычайно уязвимыми с точки зрения защищенности от умышленной или случайной порчи данных. Кроме того, поскольку операции доставки почты между пользователями в пределах одного почтового отделения выполняются исключительно средствами пользовательского агента (UA), зависание программы или компьютера на клиенте может надолго блокировать или же разрушить служебные файлы, что сделает невозможным работу других пользователей и может потребовать восстановления почтового отделения.

В более ранних версиях MTA в рассматриваемых системах функционировали исключительно под MS-DOS и требовали установки отдельного компьютера для каждого типа соединения, будь то локальная сеть, канал X.25 или коммутируемые линии. По мере развития многозадачных операционных систем сначала появилась возможность запуска старых MTA под их управлением, а затем сами MTA были переписаны как родные приложения. Примером могут служить почтовые системы MS Mail 3.5 и cc:Mail 8.0.

В настоящее время большинство производителей рассматриваемых систем переводят свои продукты в архитектуру клиент-сервер либо частично, как это сделано в Lotus cc:Mail, либо полностью, как в Microsoft Exchange.

Адресация

Системы электронной почты, такие, как MS Mail и cc:Mail, используют гораздо более простую схему адресации, нежели SMTP. MS Mail-адрес строится по следующей схеме:

NETWORK/PO/USER

где NETWORK – имя так называемой почтовой сети (или почтового домена); PO – имя почтового отделения; USER – название пользовательского почтового ящика или списка рассылки. Максимальная длина каждого из компонентов адреса не должна превышать десяти символов.

Lotus cc:Mail по сравнению с MS Mail использует более простую схему адресации, но более гибкую схему указания пользовательского почтового ящика:

USER at POST OFFICE

здесь POST OFFICE – название почтового отделения; USER – название пользовательского почтового ящика или списка рассылки, допускающее использование как псевдонима, так и реального имени. Максимальная длина адреса cc:Mail составляет 256 символов.

Поскольку рассматриваемые почтовые системы в основном предназначены для использования внутри организаций, схема назначения имен выбирается администратором произвольно. При отправке почтовых сообщений вместо реального адреса можно указывать псевдоним почтового ящика или реальное имя. Однако в любом случае перед отправкой на основе указанной информации происходит поиск реального адреса в глобальной адресной книге. Каждый почтовый ящик имеет только один адрес.

Маршрутизация

Поскольку ни MS Mail, ни cc:Mail не ориентированы на взаимодействие с какой-либо внешней службой имен, для осуществления доставки сообщений нелокальным пользователям в каждом почтовом отделении должна присутствовать статическая таблица маршрутизации. Формирование этой таблицы выполняет администратор. В случае использования синхронизации каталогов между почтовыми отделениями информация о маршрутах добавляется в таблицу автоматически.

6.4. Гибридные системы (MS Exchange Server)

Результатом накопления опыта в различных областях компьютерной индустрии стало возможным появление систем нового поколения, сочетающих в себе лучшие элементы своих предшественников и добавляющих к ним множество новых функциональных возможностей. В области электронной почты примером такой системы может служить Microsoft Exchange Server.

Общие сведения и возможности

В основу данного продукта положены, с одной стороны, удобство и простота использования, характерные для коммерческих систем, и мощные средства коммуникации, опирающиеся на общепризнанные стандарты, такие, как X.400 и SMTP, с другой. Широкий базовый набор возможностей сервера позволяет ему выполнять роль универсального связующего звена между разнородными почтовыми системами и предоставлять услуги электронной почты и групповой работы пользователям, применяющим различные протоколы доступа и клиентские программы. Так, например, пользователи cc:Mail, использующие IPX/SPX для доступа к своему серверу NetWare, могут свободно переписываться с коллегами, имеющими адреса в Internet или SPRINT. Кроме того, шлюзы сопрягаемых почтовых систем взаимно доступны в каждой из них.

Использование стандарта UNICODE на уровне хранилища позволяет поддерживать множество языков на одном сервере, а поддержка OLE-объектов – хранить и предавать любые сложные документы.

Для обеспечения прозрачной интеграции с системами на базе X.400 сервер Exchange поддерживает набор спецификаций на протокол взаимодействия между агентами передачи сообщений (MTA) и транспортные протоколы TCP, X.25 для синхронных и асинхронных линий. При пересылке сообщений через сети X.400 Exchange Server выполняет автоматическое преобразование из внутреннего формата к стандартам P2 или P22.

На уровне протокола SMTP полностью поддерживается набор стандартных и ряд расширенных (ESMTP) функций сервера, таких, как уведомление о доставке (DNR) и согласование предельного размера передаваемых сообщений (SIZE). Поддерживается маршрутизация входящей почты и фильтрация входящих соединений на основании IP-адресов. На уровне формата сообщений поддерживается UUENCODE и MIME и широкий набор национальных кодировок,

который при необходимости может быть расширен. Преобразования и перекодировки могут выполняться на основе анализа почтового адреса получателя. При соединении по SMTP серверов Exchange дополнительно можно выполнять их взаимную аутентификацию.

Прозрачная интеграция с системой MS Mail 3.x обеспечивается за счет использования метода «теневого» почтового отделения (shadow post office), подключение к которому со стороны соответствующей системы происходит стандартным образом. В случае сопряжения с Lotus cc:Mail, Exchange Server эмулирует работу MTA (cc:Mail Router) при помощи утилит EXPORT и IMPORT из стандартного комплекта почтовой системы Lotus cc:Mail.

Доступ пользователей к своим почтовым ящикам организован по принципу клиент-сервер. В качестве протоколов доступа поддерживаются:

- «родной» протокол на основе удаленного вызова процедур (RPC – Remote Procedure Calls) поверх любого транспортного протокола, поддерживаемого Windows NT;
- протокол POP3;
- протокол HTTP, через набор сценариев (ASP – Active Server Pages) сервера IIS 3.0.

Для осуществления доступа по HTTP броузер клиента должен поддерживать исполнение Java-апплетов.

Адресация в MS Exchange

В сервере Exchange использована весьма необычная на первый взгляд схема назначения адресов. Она двойная, т. е. каждый ящик или список рассылки (а если быть точным – каждый объект каталога) всегда имеет два адреса: адрес X.400 и внутренний. Во-первых, это диктовалось необходимостью обеспечения возможности использовать произвольное количество адресов различных типов для каждого почтового ящика и осуществления доставки по любому из них, что неизбежно потребовало введения параллельной адресации, не совпадающей ни с одной из существующих. Во-вторых, поскольку внутренние адреса имели собственный формат и не предназначались для применения за пределами одной организации, потребовалось наличие еще другого адреса, который мог бы быть использован для общения с внешним миром, был общеизвестен и не требовал обязательной регистрации. Этим условиям удовлетворял только адрес X.400. Чтобы обеспечить достаточную гибкость системе, ее X.400-адрес может быть динамически изменен, но не уничтожен.

Поскольку любые другие почтовые адреса, включая дополнительные X.400, являются не более чем атрибутами объекта, их набор и значение можно произвольно изменять по мере необходимости. Такой подход позволяет реализовать столь популярную сейчас концепцию плоского почтового пространства, когда упоминания о внутренней структуре организации полностью исключены из почтового адреса (например, подавляющее большинство адресов Internet в настоящее время имеет вид mailbox@company.com).

При отправке почтовых сообщений можно использовать любой из поддерживаемых типов адресов: X.400, Internet, cc:Mail и MS Mail. Установка почтовых шлюзов в другие системы позволяет вводить адреса в характерных для них форматах.

Маршрутизация в MS Exchange

Для отправки сообщений внутри почтового пространства, объединяющего серверы Exchange, системы X.400, MS Mail и cc:Mail используют статическую маршрутизацию. При отправке сообщений через сети SMTP можно использовать динамическую маршрутизацию на основе службы имен DNS и/или статическую маршрутизацию на основе таблиц. При синхронизации каталогов таблицы маршрутизации обновляются автоматически.

6.5. Почтовый каталог

Каталоги в системах электронной почты необходимы пользователю для получения расширенной информации о каком-либо предмете на основе минимального набора исходных сведений. Примером каталогов, используемых повсеместно, являются телефонные справочники. Служба каталога в той или иной мере присутствует в каждой из современных систем электронной почты, и отличия состоят лишь в том, на какие стандарты опирается та или иная реализация и какой набор сервисов представляет.

Базовые сведения о стандарте X.500

Стандарт на службу каталогов X.500 был разработан изначально для организации публичных справочников общего доступа, позволяющих хранить информацию из любой области человеческих знаний. Он представляет собой набор рекомендаций комитета CCITT/ITU, описывающих исключительно принципы построения и форматы данных для взаимодействия систем, предоставляющих сервисы поиска в глобальных хранилищах информации. Выбор средств реализации полностью возлагается на разработчика. Существуют две редакции этих рекомендаций – 1988 и 1992 гг.

Каталог (directory), построенный в соответствии с рекомендациями X.500, способен хранить информацию о наборе произвольного числа целевых объектов (objects of interest), имеющих различную структуру. Целевые объекты хранятся в *информационной базе объектов* (DIB – Directory Information Base). Каждый объект имеет связанный с ним набор сведений о структуре, свойствах и множестве разрешенных над ним действий, называемый *классом* объекта. Сами классы, в свою очередь, также трактуются как объекты.

Каждый экземпляр объекта, хранящийся в каталоге, должен соответствовать одному из зарегистрированных в DIB классов. Для обеспечения непротиворечивости данных в каталоге объекты необходимо создавать и модифицировать только в соответствии с правилами, предписанными классами этих объектов.

Для отражения того факта, что сущности реального мира могут содержать вложенные сущности и одновременно содержаться внутри других сущностей, введена *иерархия* сущностей. Сочетание информационной базы объектов и знаний об их иерархии образует *дерево информационного каталога* (DIT – Directory Information Tree). Как и положено дереву, оно имеет *корень* (root entry), *узлы*, называемые также контейнерами (container entry), и *листья* (leaf). Корень является стартовой точкой каталога. Объекты-контейнеры содержат в себе один или более объектов-листьев и/или других контейнеров. Листья не содержат вложенных объектов и, как правило, представляют собой собственно целевые объекты. Однако если объект создается «под листом», лист становится контейнером.

Набор определений и правил, регулирующих структуру информационной базы, называют *схемой каталога* (Directory Schema). Схема каталога определяет, объекты каких классов могут быть созданы в рамках каталога, каковы набор и предельные значения их атрибутов, как они могут взаимодействовать друг с другом, и где в информационном дереве каталога они могут находиться.

Внутри информационной базы каталога каждый объект должен иметь уникальное *имя* (name). Чтобы однозначно адресовать объект внутри информационной базы, его полное имя в базе также должно быть уникальным и отражать положение объекта в дереве каталога. Единственный способ получения такого имени состоит в последовательном добавлении к имени объекта имен уровней иерархии при движении вверх по дереву объектов.

Полученное имя называют *характерным именем* (DN – Distinguished Name). Имена, получаемые на промежуточных уровнях, называют *относительными характерными именами* (RDN – Relative Distinguished Name). Эти имена можно использовать при относительной адресации объектов каталога на какомлибо уровне иерархии. Строгого формата построения характерного имени именования спецификация X.500 не приводит.

Необходимо отметить, что, несмотря на некоторую схожесть формата адресов X.400 с X.500, у них совершенно разная природа и свойства. Значения ключей в адресе X.400 могут быть произвольными. В X.500 в связи с тем, что набор ключевых слов не определен стандартом, напротив, порядок следования ключей должен строго соответствовать пути к объекту в дереве каталога. В остальном адреса X.400 и X.500 вполне совместимы, и многие X.400-системы поддерживают настройки X.500 для ведения глобальных адресных книг и их автоматической репликации.

Для сокрытия внутренней структуры каталога и механизма работы с ним в составе информационной системы необходимо предусмотреть два компонента, упоминавшихся ранее: системный и пользовательский агенты каталога (DSA и DUA соответственно). При обращении клиента к каталогу за информацией об интересующих его объектах DUA выступает в роли промежуточного звена, преобразующего запрос в формат, понимаемый DSA и возвращающий полученные результаты в ожидаемом пользователем виде. В свою очередь, DSA принимает запросы со стороны пользовательских агентов и выполняет их или

переадресует запрос другим системным агентам, если запрашиваемая информация не относится к обслуживаемой им части каталога. Каталог, представляемый единым информационным пространством, на практике может быть распределен между различными DSA. В состав информационной системы входит произвольное число системных агентов, каждый из которых отвечает за различные подмножества общего информационного дерева каталога. Часть общего каталога, за обслуживание которой отвечает отдельный DSA, называют *фрагментом* (Fragment). Фрагмент включает в себя произвольное число поддеревьев из произвольных мест каталога.

Системный агент использует различную технику для обработки запросов, поступающих от пользовательского агента на те части каталога, которые не обслуживаются данным DSA:

- *цепной поиск* (chaining), когда запрос при необходимости перенаправляется другому DSA, и результаты работы последнего возвращаются пользователю;
- *перенаправление* (referral), когда системный агент инструктирует пользовательского агента, к которому DSA обратился за нужной информацией.

Использование цепного поиска и перенаправления требует возможности непосредственного взаимодействия DSA, что не всегда выполнимо, и накладывает существенные ограничения на область применения таких систем. Чтобы сократить время, затрачиваемое на обработку пользовательского запроса, применяют метод *репликации* (replication) фрагментов между системными агентами каталога. При этом DSA отслеживает изменения, вносимые в подотчетный ему фрагмент, и доставляет их остальным системным агентам. В этом случае относительно актуальная копия всего каталога доступна для поиска каждому DSA системы, однако использование такой схемы требует дополнительных затрат ресурсов на размещение избыточных копий информации. Для почтовых систем данный вариант организации доступа к каталогу единственно возможный, так как отдельные фрагменты могут не иметь непосредственного соединения друг с другом. Единицей репликации данных является *пространство имен* (Name Context), представляющее собой отдельную ветвь общего дерева.

Несмотря на массу достоинств, реальных систем, полностью отвечающих рекомендациям X.500, не так много, и все они, как правило, функционируют либо на уровне региональных административных доменов, либо в государственных учреждениях и силовом секторе. Высокая сложность реализации и громоздкость интерфейсов взаимодействия подсистем привели к появлению параллельных служб каталогов, заимствующих идею X.500, но по-другому реализующих протоколы доступа и форматы передачи данных.

Каталоги частных систем

В терминах X.500 в частных системах реализована схема с репликацией отдельных фрагментов каталога между системными агентами почтовых от-

делений. Сам каталог ограничен малым числом уровней иерархии (тремя для MS Mail и двумя для cc:Mail). База объектов содержит небольшое число классов, таких, как почтовый ящик, список рассылки, общая папка, шаблон, таблица маршрутов, внешний адресат и почтовый шлюз. Шаблоны позволяют модифицировать набор атрибутов почтового ящика и списков рассылки. Создание новых классов объектов в базе объектов не предусмотрено.

В качестве информационной базы глобального каталога выступает глобальная адресная книга, содержащая данные об иерархии организации и пользователях в ее составе. Фрагментом в данном случае является локальное почтовое отделение. Поскольку данные системы используют файловый доступ для выполнения всех операций, пользовательский агент каталога интегрирован с почтовым агентом и выполняет роли как DUA, так и DSA при поиске информации в глобальной адресной книге. По той же причине при сборе изменений о подотчетном фрагменте каталога в роли DSA выступает внешняя программа, которую запускают на отдельном компьютере. Эта программа формирует файл изменений для локального почтового отделения. Собственно репликация выполняется путем пересылки изменений каталога в виде письма выделенному серверу каталога, имеющего специальный почтовый адрес. В задачи упомянутого сервера входят слияние изменений ото всех почтовых отделений, обновление адресной книги и рассылка модификаций к текущей адресной книге системному агенту каждого отделения. На основе полученного файла модификаций при следующем запуске локальный DSA вносит изменения в глобальную адресную книгу, затем снова контролирует изменения в структуре локального отделения и при необходимости создает новый файл изменений, направляемый опять же серверу каталога. После чего процесс повторяется.

Несмотря на кажущуюся громоздкость, такая схема обеспечивает достаточно высокую эффективность ведения общего адресного списка и способна обслуживать до 500000 почтовых абонентов.

Каталог *Exchange*, связь с каталогом X.500

Каталог сервера Exchange хотя и основан на спецификациях X.500, не следует им в области использования протоколов передачи данных и двоичного формата потоков данных. Однако с точки зрения реализации объектного хранилища и разделения функций между DSA и DUA, Exchange полностью следует рекомендациям X.500. Информационное дерево каталога (рис. 6.1) Exchange Server содержит все компоненты классического каталога, включая корень, контейнеры, листья и схему данных.

Каждый объект каталога имеет уникальное имя в каталоге, полное и относительное характерное имена. Характерные имена объектов, таких, как пользовательские ящики, списки рассылки и т. д., можно использовать в качестве их почтового адреса во внутреннем формате Exchange. Следует, однако, помнить, что внутренние адреса имеют силу только в том случае, если адресуемый объект расположен в пределах той же организации, что и отправитель.

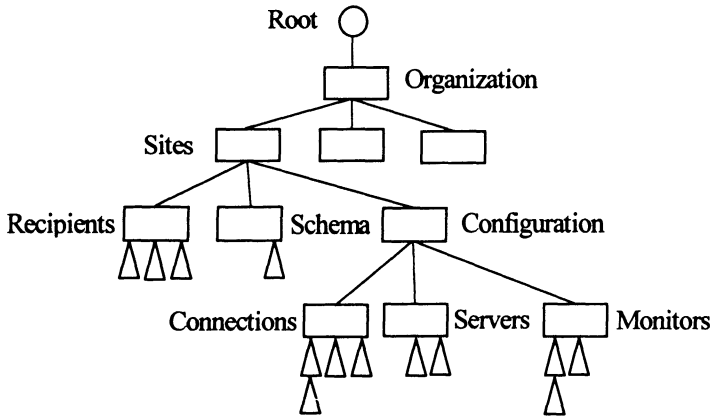


Рис. 6.1. Информационное дерево каталога Exchange Server

Exchange использует метод репликации фрагментов каталога, т. е. каждый сервер хранит локальную копию каталога организации. Запросы от пользовательских агентов каталога обрабатываются локально во всех случаях, кроме обращений к общим папкам. Если сервер не имеет на себе запрошенной копии, он на основании данных каталога переадресует клиента к DSA сервера, на котором копия папки присутствует.

Каждый сервер обслуживает фрагмент, состоящий из четырех неперекрывающихся пространств именований: организация (Organization), площадки (Sites), настройка (Configuration) и схема каталога (Schema).

Облегченный протокол доступа к каталогу (LDAP)

Облегченный протокол доступа к каталогу (LDAP – Lightweight Directory Access Protocol) был создан для обеспечения работы «легких» пользовательских агентов, таких, как Internet-броузеры, с каталогами, использующими архитектуру X.500. Данный протокол рассчитан исключительно на использование поверх TCP/IP и использует упрощенный набор команд для общения клиента с сервером. Согласно спецификации на протокол, с его помощью можно выполнять операции чтения, поиска, сравнения и обновления данных в каталоге. Принятая в LDAP схема проверки полномочий на основе единственной текстовой строки в открытом виде и отсутствие какой бы то ни было поддержки назначения прав доступа на отдельные элементы каталога ограничивают реальную сферу применения данного протокола областью справочников общего доступа, допускающих работу исключительно анонимных пользователей. Конкретные реализации протокола могут отличаться, например, поддержкой шифрования трафика по SSL 3.0 или проверкой права на установление соединения на основе имени и пароля в операционной системе.

6.6. Службы совместного использования информации

Чтение новостей и ведение дискуссий всегда было очень популярным занятием. Системы электронной почты вывели это общечеловеческое хобби на качественно новый уровень. Сегодня практически любая почтовая система предоставляет возможности ведения дискуссий и обмена информацией между ее пользователями. Рассмотрим некоторые из них.

Служба новостей Internet

Служба новостей (Network News) была создана в начале 80-х годов для организации электронных досок объявлений пользователей систем UNIX. Она изначально была ориентирована на работу в архитектуре клиент-сервер и позволяла вести дискуссионные группы, распределенные между несколькими серверами с возможностью автоматической репликации вновь поступающих сообщений. Для общения серверов между собой и клиента с сервером был создан протокол передачи сетевых новостей (NNTP – Network News Transport Protocol), который в несколько модифицированном виде успешно используют по сей день. В начальных реализациях служба имела некоторые недостатки, в частности, плохую защиту от возникновения бесконечных циклов передачи сообщений от сервера к серверу, не совместимый с почтовыми системами формат данных и адресации и т. п. Для устранения недостатков первых версий NNTP в лаборатории AT&T была разработана служба USENET, впоследствии ставшая общепризнанным стандартом и успешно существующая и сегодня. Используя тот же протокол NNTP, что и предшествующие реализации, USENET ввела в употребление новый формат и способ адресации сообщений, совпадающий с принятыми в SMTP-системах. Информация, специфическая для службы новостей, указывалась в расширенных полях заголовка сообщения. Это, в частности, позволило разрабатывать клиентские программы для чтения почты и новостей на основе единого кода, а также использовать существующие сети SMTP для получения информации в тех местах, где непосредственный доступ к серверу новостей по каким-либо причинам был невозможен. Кроме того, было введено понятие контрольных сообщений, предназначенных для обмена управляющей информацией между серверами новостей и упрощающих процесс автоматического создания и удаления дискуссионных групп и ликвидации устаревших сообщений.

Вся информация, хранимая в USENET, представляется единым иерархическим деревом, организованным по тематическому признаку. В этом смысле USENET выступает в роли тематического каталога, содержащего мнения людей на ту или иную тему. Сообщения (*статьи* – articles), объединенные общей тематикой, помещаются в тематические группы, называемые *группами новостей* (newsgroups). Группы новостей, в свою очередь, могут входить в другие группы, образовывая тематические *иерархии*. Каждый уровень иерархии называется *категорией*. В рамках категории группа имеет уникальное имя. Полное характерное имя группы получается последовательным добавлением

слева направо имен категорий при движении вниз от корня по дереву иерархии. Имена категорий разделены точкой.

Иерархии, или их отдельные ветви, реплицируются между серверами, образующими пространство USENET. В качестве единицы репликации выступает отдельная статья. При репликации использована схема издатель – подписчик. Каждый сервер USENET может быть *подписан* на некоторое подмножество групп, предоставляемых другими серверами. Одновременно он может *публиковать* некоторое подмножество групп, расположенных непосредственно на нем, в том числе группы, получаемые по подписке. В терминах USENET репликация именуется *заполнением* (*feed*). В зависимости от того, какой сервер выступает инициатором этого процесса, различают два типа заполнения:

вытягиванием (pull feed), когда сервер, ожидающий поступления новых статей, сам обращается к своему издателю;

проталкиванием (push feed), когда сервер, имеющий у себя новые статьи, пытается передать их подписчику.

Еще одним преимуществом службы USENET является возможность создания *модерированных* групп новостей. В модерируемой группе каждое новое сообщение автоматически перенаправляется лицу, выполняющему роль цензора или *модератора*. Если сообщение не противоречит уставу конференции и «одобрено» модератором, оно становится публично доступным для прочтения. В противном случае – просто удаляется.

Поскольку служба новостей изначально создавалась как средство ведения хранилища публично доступной информации, в ней не был предусмотрен механизм назначения и проверки прав доступа на отдельные ветви каталога. Большинство существующих служб новостей способны выполнить лишь однократную проверку имени и пароля пользователя при установлении соединения с сервером, после удачного завершения которой все конференции становятся доступны клиенту. Кроме того, не предусмотрена возможность авторизации серверов и контрольных сообщений. Как следствие этого, массовое применение USENET оправдано только для организации публичных групп новостей с анонимным режимом доступа.

Простота и одновременно высокая эффективность реализации распределенного доступа к данным и широкая доступность обеспечили службе новостей Internet огромную популярность. Без преувеличения можно сказать, что по популярности этот сервис всемирной сети занимает одно из первых мест. Практически все поставщики систем электронной почты либо уже реализовали, либо планируют непосредственную поддержку службы новостей в своих продуктах.

Службы частных систем

Службы поддержки совместного использования информации и ведения дискуссий давно стали обязательной составляющей в частных системах электронной почты. В MS Mail такая служба носит название *общих папок* (shared folders), в cc:Mail – *досок объявлений* (bulletin board). Эти службы ориентиро-

ваны в большей мере на организацию бизнес-процессов в рамках рабочей группы и призваны решать другие задачи, отличные от организации дискуссий мирового масштаба, как, например, службы новостей Internet.

В частных системах изначально заложена возможность хранения информации, имеющей нетекстовую структуру, такую, например, как документы и/или двоичные файлы (хотя в настоящее время это отличие уже не столь заметно). Наряду с поддержкой иерархий произвольной вложенности, эти системы позволяют более или менее гибко назначать права на ресурсы совместного пользования отдельным пользователям в рамках почтового отделения. Среди стандартных прав присутствуют чтение сообщений в папке, создание, модификация и удаление сообщений. Существует возможность назначить владельца папки, который будет назначать права доступа другим пользователям почтового отделения. Все действия по созданию и администрированию общих папок выполняют непосредственно из клиентских программ, так как поддержка этих функций интегрирована в последние.

Однако службы совместного использования имеют один значительный недостаток, они, как правило, ограничены рамками одного почтового отделения. Каждый разделяемый ресурс имеет свой почтовый адрес в глобальной книге организации, позволяющий нелокальным пользователям посылать сообщения в дискуссионные группы, но этот способ имеет ограниченную область применения. Последние версии cc:Mail поддерживают репликацию досок объявлений между почтовыми отделениями в рамках процесса репликации каталога, в MS Mail обеспечение аналогичных возможностей требует применения расширений сторонних производителей.

Службы Exchange

В терминологии сервера Exchange, подобно MS Mail, для определения ресурсов совместного использования применяют термин *общие папки* (public folders). Однако помимо названия и отдаленного внешнего сходства в окне клиента между ними нет ничего общего.

Следуя терминологии X.500, общие папки Exchange являются фрагментом дерева глобального каталога системы и, по определению, обладают всей свойственной ему функциональностью. Каждая папка является контейнером и может содержать произвольное количество вложенных контейнеров и сообщений. Общие папки поддерживают гибкую систему назначения прав на выполнение пользователями таких операций, как поиск, создание, модификация и удаление папок и сообщений. Назначение прав на общие папки выполняется на основе глобальной адресной книги.

Как и положено фрагменту каталога, различные его части, или пространства имен, представляющие собой папки различных уровней с вложенными в нее объектами, могут реплицироваться между серверами Exchange, относящимися к одному каталогу. Нелокальные пользователи могут выполнять все

предписанные им действия над реплицированной на их сервер копией. При этом будет обеспечена автоматическая репликация внесенных изменений на все существующие копии модифицированной информации.

Поскольку каждый объект в каталоге Exchange может иметь одновременно несколько почтовых адресов различных типов (SMTP, X.400, cc:Mail, MS Mail), пользователи внешних организаций и почтовых систем могут направлять сообщения непосредственно в общие папки по электронной почте. Это, в частности, позволяет включать общие папки в списки рассылки.

Сведения об иерархии общих папок хранятся в каталоге и автоматически реплицируются на все серверы организации. Репликация же самих данных настраивается и выполняется отдельно. Так как схема репликации папок выбирается произвольно, возможна ситуация, когда конкретный сервер не имеет локальной копии запрашиваемой пользовательским агентом каталога папки. В этом случае Exchange использует механизм переадресации запроса системному агенту сервера, на котором, согласно данным из каталога, данная папка, или ее копия, должна присутствовать. Применение механизма переадресации позволяет избежать хранения всех данных на каждом сервере, что в целом повышает общую производительность почтовой системы в конфигурациях с большим числом клиентов, перекладывая нагрузку по обслуживанию общих папок на выделенные серверы с достаточным запасом ресурсов.

Информационное хранилище Exchange поддерживает объектную модель OLE, что позволяет сохранять документ из OLE-совместимых приложений непосредственно в общих папках и проводить операции поиска и сортировки по стандартным и определяемым пользователем свойствам документа, например, автор, ключевые слова, число страниц и т. п.

Еще одним немаловажным достоинством общих папок Exchange является возможность организации репликации с серверами службы новостей USENET. Специально для этой цели в Exchange Server 5.0 предусмотрена возможность создания моделированных папок и реализована поддержка контрольных сообщений USENET на создание и удаление групп новостей и удаление отдельных сообщений. Exchange Server поддерживает оба типа наполнения: путем вытягивания (pull feed) и проталкивания (push feed). Имеется возможность принимать не все предлагаемые сервером USENET группы новостей, а только некоторые. Принимаемые конференции сети USENET помещаются напрямую в общие папки, и далее эта информация реплицируется обычным образом на остальные серверы Exchange. Пользователи этих серверов могут работать с группами новостей как с обычными папками, просматривая, создавая собственные или отвечая на существующие сообщения. По завершении обратной репликации результаты их работы, в конечном итоге, попадают в группы новостей USENET. Таким образом, пользователям, не имеющим прямого доступа в Internet, обеспечена возможность принимать участие в дискуссиях на интересующие их темы.

Дополнительно к поддержке взаимодействия на уровне сервер-сервер сети USENET, Exchange может выступать в роли сервера новостей для обычных news-клиентов, использующих программы просмотра новостей Internet, что позволяет использовать один сервер Exchange как для организации почты внутри организации, так и поддержки собственных групп новостей в Internet. Отличительной особенностью сервера новостей на Exchange является возможность назначения прав пользователям на работу с той или иной конференцией.

7. СЕТЕВАЯ ОС NETWARE

В данной главе рассмотрены назначение и основные возможности сетевой ОС NetWare, *ее основные достоинства и недостатки. Приведены примеры использования ОС при построении различных типов серверов, рабочих станций и коммуникационных узлов, а также примеры приложений на серверах* NetWare.

7.1. Назначение и основные возможности ОС NetWare

Назначение ОС NetWare

Как известно, сеть – это совокупность компьютеров, связанных друг с другом для обеспечения обмена данными и совместного использования различных ресурсов. Сетевыми ресурсами являются принтеры, файлы, прикладные программы и т. д.

Для управления сетью разрабатывают специальные сетевые операционные системы (ОС), которые по своей организации можно разделить на одноранговые (Peer-To-Peer Network) и с выделенным файловым сервером (Dedicated File Server Network).

В одноранговых сетях на каждой рабочей станции (компьютере) сети могут быть загружены две группы модулей: *сервера и клиента* (рис. 7.1).

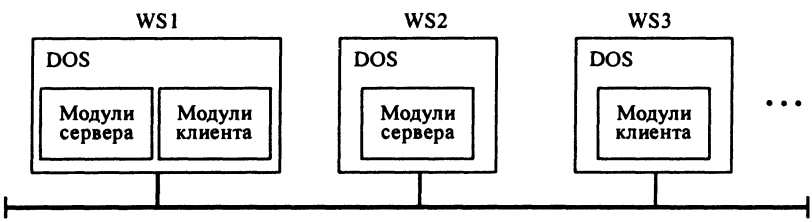


Рис. 7.1. Пример загрузки модулей одноранговой сети

Загрузка в оперативную память (ОП) рабочей станции модулей сервера обеспечивает доступ других пользователей к ресурсам этого компьютера. А наличие модулей клиента позволяет пользователю иметь доступ к ресурсам других рабочих станций сети. Указанные группы программ могут использоваться в различных сочетаниях. Так, рабочие станции (WorkStation) WS1 и WS2 (см. рис. 7.1) могут выступать в качестве серверов, т. е. их ресурсами могут пользоваться другие станции, а станции WS1 и WS3 имеют доступ к другим компьютерам сети.

К одноранговым сетям относят следующие сетевые операционные системы:

- NetWare Lite, Personal NetWare (Novell);
- Windows For Workgroups (Microsoft);
- LANtastic (Artisoft).

Здесь в скобках указаны названия фирм-изготовителей соответствующих продуктов. В табл. 7.1 перечислены преимущества и недостатки одноранговых ОС.

Таблица 7.1. **Преимущества и недостатки одноранговых ОС**

Преимущества	Недостатки
Простота инсталляции Обеспечивают доступ к ресурсам других рабочих станций	Низкая производительность сети, что объясняется небольшой мощностью рабочих станций Ограниченные возможности по обеспечению связи удаленных сегментов сети Отсутствие развитых средств управления сетью Ограниченные возможности по обеспечению режима работы СУБД «клиент-сервер»

В сетях с выделенным сервером сетевая ОС инсталлируется и загружается на отдельной станции, называемой *файловым сервером* (File Server). Рабочие станции имеют доступ к общим данным и другим ресурсам, хранящимся на файловом сервере (рис. 7.2).

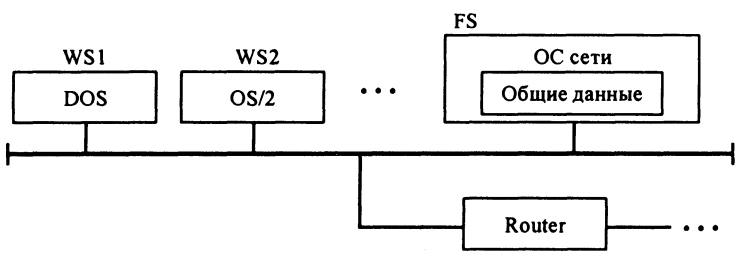


Рис. 7.2. Пример сегмента сети с выделенным сервером

К операционным системам, устанавливаемым на файловом сервере, относятся следующие ОС:

- Vines 5.53 (Banyan);
- OS/2 LAN Server 4.0 Advanced (IBM);
- Windows NT Server 4.0 (Microsoft);
- NetWare 3.x, 4.x, 5.x (Novell).

Рабочие станции могут функционировать под управлением различных ОС: MS DOS, OS/2, UNIX, Macintosh, Windows NT Workstation.

В табл. 7.2 перечислены основные преимущества и недостатки ОС с выделенным сервером.

Таблица 7.2. **Преимущества и недостатки ОС с выделенным сервером**

Преимущества	Недостатки
Высокая производительность сети за счет использования файлового сервера большой мощности Развитые аппаратные и программные средства связи удаленных сегментов сети и рабочих станций Развитые средства управления и администрирования в сети Широкие возможности режима работы СУБД «клиент-сервер»	Некоторая сложность в освоении Ограниченные возможности доступа к ресурсам рабочих станций

Чтобы устранить недостатки, присущие сетям рассмотренных типов, часто на одном сегменте сети устанавливают две ОС: одноранговую и с выделенным сервером. Наблюдается некоторое сближение сетей различных типов. Например, для увеличения производительности одноранговой сети LANtastic фирма Artisoft выпустила сервер CorStream, функционирующий под управлением NetWare 4.01.

Признанными лидерами сетевых ОС с выделенным сервером являются Windows NT Server 4.0 и NetWare 3.x,4.x,5.x. ОС NetWare версий 3.x, 4.x и 5.x предназначены для обеспечения доступа к общим ресурсам сети со стороны нескольких пользователей. В качестве таких ресурсов выступают файлы данных, принтеры, модемы, модули и т. д. Для комплексного решения этой проблемы потребовались большие усилия со стороны различных фирм. В процессе разработки ОС NetWare были решены некоторые важные задачи, которые рассматриваются в следующих пунктах пособия.

Возможности ОС NetWare, предоставляемые пользователю

ОС NetWare предоставляет пользователям следующие возможности:

- поддерживает коллективное использование файлов;
- обеспечивает доступ к сетевым принтерам;
- предлагает средства для работы с электронной почтой;
- поддерживает работу СУБД различных типов;

• обеспечивает доступ к файловому серверу со стороны рабочих станций, функционирующих под управлением различных операционных систем;
• предлагает средства, позволяющие объединять удаленные сегменты сети;
• обеспечивает «прозрачность» доступа локальных и удаленных пользователей к ресурсам сети;
• предлагает средства для надежного хранения данных;
• обеспечивает защиту ресурсов сети от несанкционированного доступа;
• поддерживает динамически расширяемые многосегментные тома на нескольких дисках файлового сервера;
• предоставляет средства управления ресурсами корпоративных сетей: единый каталог сетевых ресурсов NDS в NetWare 4.x/5.0;
• обеспечивает передачу и обработку данных с использованием разных протоколов: SPX/IPX, TCP/IP, NetBIOS, AppleTalk;
• поддерживает работу суперсерверов в симметричном режиме функционирования (ОС NetWare 4.1 SMP, в NetWare 5.0 работа суперсерверов поддерживается на уровне ядра ОС).

Рассмотрим некоторые возможности более подробно.

Прежде всего ОС NetWare позволяет пользователям обращаться к общим файлам, хранящимся на файловом сервере. Это, с одной стороны, позволяет не дублировать общие данные на рабочих станциях, а, с другой стороны, обеспечивает взаимодействие пользователей через файловый сервер.

Если всем пользователям сети необходимо выводить данные на печать, а число принтеров меньше, чем число рабочих станций, то NetWare позволяет сделать печатающие устройства разделяемыми, т. е. доступными всем клиентам сети.

Под управлением NetWare функционирует шлюз электронной почты (ЭП) MHS (Message Handling Service), состоящий из NLM-модулей. Он вошел в состав NetWare 4.1 как штатное средство. MHS управляет сбором, маршрутизацией и доставкой сообщений через разнородные шлюзы ЭП и линии связи. С MHS совместимо более 200 пакетов электронной почты, функционирующие на рабочих станциях.

ОС NetWare обеспечивает работу СУБД, поддерживающих две технологии обработки запросов: «клиент-файл» и «клиент-сервер». В СУБД типа «клиент-файл» запросы прикладной программы к базе данных (БД) транслируются и выполняются на рабочей станции, на файловом сервере хранятся файлы БД, индексные и технологические файлы. К этому классу СУБД относятся следующие популярные пакеты: dBase (Borland), Clipper (Computer Association), FoxPro (Microsoft), Paradox (Borland), Clarion (Clarion) и т. д. В СУБД типа «клиент-сервер» запрос к БД передается серверу СУБД, функционирующему как совокупность NLM-модулей на файловом сервере. На сервере запрос транслируется и выполняется. На рабочую станцию возвращаются только результаты выполнения запроса. Для СУБД этого типа характерно уменьшение сетевого трафика и увеличение нагрузочной способности сети. СУБД Oracle 7/8 (Oracle), SQLBase (Centura), Btrieve (Btrieve Technologies), Progress 7 (Progress Software) и другие поддерживают технологию «клиент-сервер».

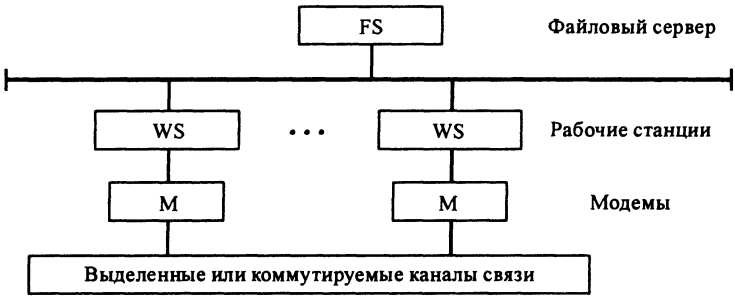


Рис. 7.3. Случай выделения модема каждой станции

В ОС NetWare обеспечена возможность доступа к файловому серверу со стороны рабочих станций, функционирующих под управлением различных ОС: MS DOS, OS/2, UNIX, Macintosh, Windows NT Workstation и т. д. В настоящее время разработаны средства взаимодействия NetWare с другими сетевыми ОС: OS/2 LAN Server, Windows NT Server, UNIX.

Фирма Novell разработала программное обеспечение выделенных маршрутизаторов, позволяющих объединять удаленные сегменты сети, например продукт NetWare Multi Protocol Router 3.x.

Если каждому пользователю сети необходимо обеспечить доступ к удаленным ресурсам (шлюзам электронной почты, удаленным сегментам, удаленным рабочим станциям и т. д.), то это можно осуществить двумя способами: выделить каждой станции свой модем (рис. 7.3) или сделать модемы разделяемыми (общими). В последнем случае модемы необходимо подключить к файловому серверу и использовать программные продукты (группы NLM-модулей) NACS, NetWare Connect (3.x, 4.1) или Novell Internet Access Server 4.x (4.11, 5.0) (рис. 7.4).

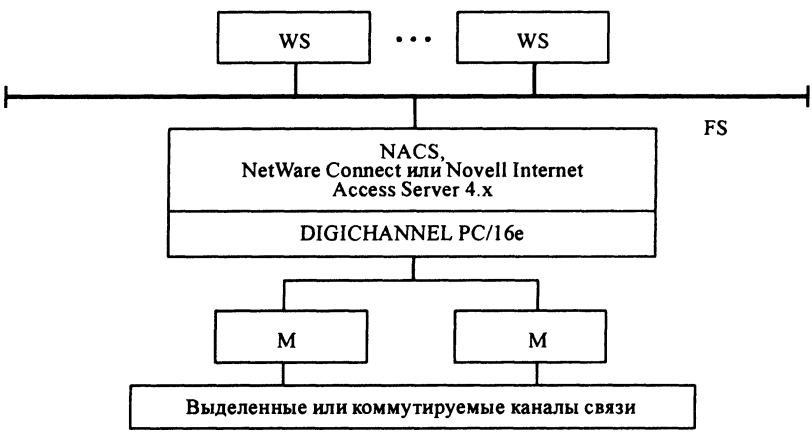


Рис. 7.4. Разделяемый модем

Иногда рабочую станцию подключают к сегменту сети, используя выделенные или коммутируемые каналы связи. В этом случае говорят, что рабочая станция является удаленной. Для обеспечения прозрачного доступа удаленной станции к ресурсам файлового сервера необходимо было решить ряд важных проблем. При активизации приложения на локальной рабочей станции оно, как правило, считывается в оперативную память этого компьютера и там выполняется. Такая схема загрузки на удаленной рабочей станции может привести к резкому увеличению трафика каналов связи. Чтобы предотвратить передачу программ по каналам связи, на выделенном локальном сервере устанавливают многозадачные среды, например, DESQview386 или WinFrame (Citrix). В этих средах и выполняются утилиты и приложения DOS (800 кб ОП на один сеанс) или Windows, инициируемые пользователями удаленных рабочих станций. По каналам связи передаются только данные, вводимые или выводимые на экраны рабочих станций, функционирующих в режиме терминала (рис. 7.5).

Для выполнения Windows-приложений, инициируемых с удаленных рабочих станций, можно также использовать и более старую многозадачную среду WinView for Networks v.2.3 (Citrix System Inc.), разработанную на базе OS/2 (5 или 10 сеансов Windows, 4 Мб ОП на один сеанс плюс 8 Мб под WinView).

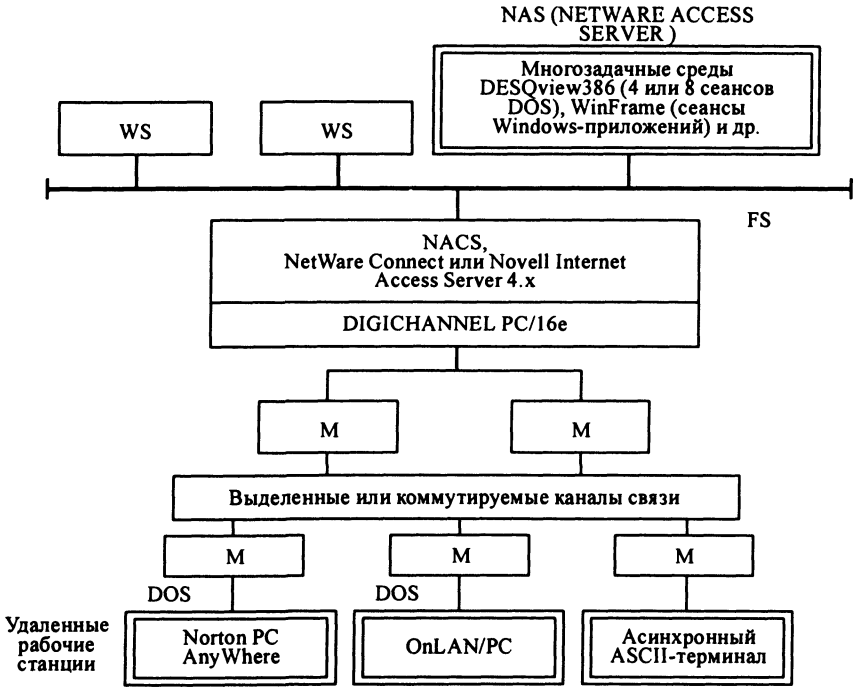


Рис. 7.5. Схема подключения удаленных рабочих станций

ОС NetWare поддерживает несколько уровней SFT, обеспечивающих надежное хранение данных:

- на уровне SFT-I:
  дублирование таблиц DET и FAT тома,
  проверка записи на диск последующим чтением,
  динамическая переадресация блоков (Hot Fix);
- на уровне SFT-II:
  зеркальное отображение дисков (Disk Mirroring),
  дублирование дисков (Disk Duplexing),
  система отслеживания транзакций TTS (Transaction Tracking System),
  использование устройств бесперебойного питания (UPS);
- на уровне SFT-III:
  зеркальное отображение файловых серверов.

Возможности ОС NetWare по администрированию

Администратор сети – специалист, в круг обязанностей которого входит выполнение следующих основных функций:

- инсталляция операционной системы NetWare;
- описание информационной среды;
- настройка операционной среды;
- настройка сетевой печати;
- мониторинг сети и управление сетевыми ресурсами;
- архивирование и восстановление данных сети.

В процессе инсталляции ОС NetWare администратор должен ответить на ряд вопросов в диалоговом режиме. Инсталляция NetWare 3.x выполняется с дискет, а NetWare 4.x/5.x – с CD-ROM.

Описание информационной среды включает выполнение следующих действий:
описание сетевых ресурсов: объектов пользователей, групп и т. д.,
создание каталогов и файлов пользователей,
назначение опекунских прав по отношению к каталогам и файлам,
назначение атрибутов каталогам и файлам.

Описание объектов, назначение опекунских прав и атрибутов выполняет администратор с помощью утилит NetWare. Каталоги и файлы могут быть созданы на сервере NetWare средствами ОС рабочей станции. Следует подчеркнуть, что в NetWare 4.x/5.x основным средством администрирования в среде Windows является программа NetWare Administrator (для 4.x – NWADMIN.EXE, для 5.x – NWADMN32.EXE).

Настройка операционной среды, выполняемая с помощью утилит NetWare, включает:
инсталляцию конфигурационных файлов рабочих станций;
разработку системных и пользовательских процедур подключения к сети;
описание меню пользователей или управление рабочими станциями.

Настройка сетевой печати предполагает описание:
объектов печати (очередей, принтеров, серверов печати);
форм печати (размера бумаги);
конфигураций заданий на печать.
Сетевая печать также настраивается с помощью утилит.

Мониторинг сети – оценка ее характеристик производительности и надежности. В ОС NetWare мониторинг осуществляется с помощью программы файлового сервера MONITOR.NLM и некоторых других утилит, запускаемых на рабочей станции. Но возможности этих средств весьма ограничены. Гораздо большими возможностями обладают специально разработанные для этой цели пакеты, например NMS (Novell), ManageWise (Novell, Intel). Программный продукт ManageWise позволяет реализовать следующие функции:

- воспроизведение топологии сети и инвентаризацию оборудования;
- мониторинг файловых серверов и сегментов сети;
- управление файловым сервером (средствами удаленной консоли), концентраторами, рабочими станциями.

Более подробно эти вопросы рассмотрены в следующих разделах.

Возможности ОС NetWare, предоставляемые программисту

Разработчик ОС NetWare фирма Novell поставляет ряд API-интерфейсов, позволяющих получить доступ к сетевым средствам из прикладной программы, а именно:

- API-интерфейсы для прикладных программ, выполняемых на рабочих станциях;
- API-интерфейсы для прикладных программ, выполняемых на файловом сервере (для NLM-модулей).

API-интерфейсы для рабочих станций включают:
С-библиотеку NetWare для DOS;
описание системных вызовов NetWare для языка Ассемблер;
справочник программиста по OS/2;
справочное руководство по Macintosh в NetWare.

С-библиотека NetWare для DOS представляет собой библиотеку функций, разработанных фирмой Novell для того, чтобы программисты на языке С могли реализовать доступ к службам NetWare в программах, выполняемых под управлением DOS. Эти функции включают в библиотеки для С-компиляторов фирм Watcom, Microsoft, Borland и Lattice. Хотя функции создаются только для языка С, можно создать интерфейсы и для других языков программирования, используя описание системных вызовов NetWare для языка Ассемблер.

NLM-модули выполняются на файловом сервере и разрабатываются, как правило, с помощью компилятора Watcom С. API-интерфейсы для этих модулей представляют собой библиотеки, которые имеют форму загружаемых NLM-модулей (CLIB.NLM, DSAPI.NLM и т. д.). Например, библиотека CLIB.NLM содержит почти все функции, представленные в С-библиотеках NetWare для

DOS. Связь между каким-либо NLM-модулем и требуемыми функциями библиотеки выполняется динамически при загрузке этого NLM-модуля в оперативную память файлового сервера.

В NetWare 4.11 для разработки серверных приложений дополнительно используется язык NetBasic, а в NetWare 5.0 – еще и язык Java.

7.2. Достоинства и недостатки

Достоинства ОС NetWare

ОС NetWare имеет некоторые преимущества по сравнению со своим конкурентом Windows NT Server 4.0. Это – маршрутизация всех протоколов на сервере, удаленное управление самим сервером и эффективная поддержка клиентских станций, работающих под управлением DOS.

Более того, NetWare 4.11/5.0 обеспечивает поддержку ряда служб, необходимых для ОС сети масштаба предприятия. Например, служба справочника NetWare (NDS) стала мощным средством управления большими корпоративными сетями. Система безопасности включает функцию ограничения занимаемого на томе сервера объема данных пользователя. Производительность ОС NetWare при операциях с файлами и службами сетевой печати по-прежнему выше, чем у ее конкурентов, особенно в случае крупных разнородных сетей. По сравнению с другими сетевыми ОС, NetWare поддерживает бóльшее число операционных сред клиентов, включая DOS, Windows, OS/2, Macintosh и UNIX.

ОС NetWare лидирует и по числу представленных на рынке аппаратных и программных средств третьих производителей, которые расширяют и дополняют ее функциональные возможности. Система имеет больше возможностей для резервного копирования и хранения данных, больше управляющих утилит и сетевых приложений, чем любая другая ОС. Еще одно преимущество NetWare заключается в том, что для ее обслуживания легче найти квалифицированный обслуживающий персонал.

Характеристики ОС NetWare представлены в табл 7.3.

Таблица 7.3. **Характеристики ОС NetWare**

Характеристика	Значение
Минимальный объем ОП файлового сервера	Для NetWare 4.x – 9 Мб (Windows NT Server 3.51 – 16 Мб, OS/2 Lan Server 4.0 Advanced – 14 Мб)
Поддержка клиентских платформ	DOS, Windows, WFW, UNIX, OS/2, Macintosh, Windows NT Workstation
Регистрация пользователей и доступ к сетевым ресурсам	В NetWare 4.x поддерживается однократная регистрация при входе в сеть
Почтовые шлюзы, включенные в поставку ОС	В NetWare 4.1 включен шлюз MHS (в ОС Windows NT Server 3.51 и OS/2 Lan Server 4.0 Advanced шлюзы не входят в базовую поставку)

Окончание табл. 7.3

Характеристика	Значение
Средства маршрутизации, включенные в поставку	IPX, TCP/IP, AppleTalk (в ОС Windows NT Server 3.51 и OS/2 Lan Server 4.0 Advanced – только TCP/IP)
Число прав доступа к файлам и каталогам	8 (Windows NT Server 3.51 – 6, OS/2 Lan Server 4.0 Advanced – 7)
Поддержка компрессии файлов на жестком диске	Поддерживается в NetWare 4.x (в OS/2 Lan Server 4.0 Advanced такие средства отсутствуют)
Время считывания 7 Мб с файлового сервера (клиенты – WFW; сервер – 486/DX2, 32 Мб, SCSI, 1 Гб; шина – витая пара на 10 Мбит/с)	NetWare 4.1 – 0,86 мин, Windows NT Server 3.51 – 1,17 мин, OS/2 Lan Server 4.0 Advanced – 1,55 мин
Время поиска по базе данных Lotus Notes (клиенты – WFW; сервер – 486/DX2, 32 Мб, SCSI, 1 Гб; шина – витая пара на 10 Мбит/с)	NetWare 4.1 – 1,17 мин, Windows NT Server 3.51 – 2,67 мин, OS/2 Lan Server 4.0 Advanced – 9,83 мин

Ориентация ОС NetWare на корпоративные сети

В настоящее время много публикаций посвящено компьютерным сетям масштаба предприятия. Так выглядит дословный перевод термина «EnterpriseWide Networks», получившего распространение в западной литературе. В отечественных материалах их чаще называют корпоративными сетями.

Отличительными характеристиками, наличие которых позволяет считать сеть корпоративной, являются:

- большое количество объединенных в общую сеть компьютеров, в том числе большое число файловых серверов, серверов баз данных, приложений и т. д.;
- гетерогенный характер сети – различные протоколы, разнородные среды передачи, произведенные разными компаниями компьютерные платформы, различные операционные системы;
- наличие нескольких локальных вычислительных сетей, территориально отстоящих друг от друга.

При переходе от локальной сети к корпоративной необходимо решить следующие задачи:

1) объединить различные компьютерные платформы в единую сеть;

2) реализовать поддержку маршрутизации различных сетевых протоколов;

3) объединить удаленные локальные сети с помощью мостов, маршрутизаторов и шлюзов;

4) организовать доступ большого числа пользователей к ресурсам единой сети и управление ресурсами.

Важность решения первой задачи связана с тем, что каждая из платформ имеет сильные и слабые стороны. Novell предлагает следующее распределение ролей между платформами:

операционная система NetWare – файловый, почтовый (MHS) и коммуникационный (NetWare Connect, Novell Internet Access Server 4.x) серверы,

операционная система UnixWare – сервер баз данных и приложений.

При решении 2-й и 3-й задач можно использовать следующие продукты фирмы Novell: NetWare Multi Protocol Router v.3.x (MPR), NetWare/IP, NetWare FLeX/IP 1.2c, NetWare NFS 1.2c.

Используя MPR, можно построить глобальную разнородную сеть на базе NetWare и PC-совместимых компьютеров. В одну сеть могут быть объединены сети, работающие по протоколам IPX, IP, AppleTalk, Novell NetBIOS, OSI, FTAM с использованием физических сред передачи Ethernet, Token Ring, ARCnet, FDDI, LocalTalk.

Продукт NetWare/IP обеспечивает полную интеграцию сетей NetWare в среду протокола TCP/IP. Он позволяет пользователям сетей, работающих по протоколу TCP/IP, использовать сети NetWare и приложения для них.

Пакеты NetWare FLeX/IP 1.2c и NetWare NFS 1.2c предназначены для использования в сетях, где необходим доступ пользователей UNIX к ресурсам NetWare (принтерам и файлам).

Четвертая задача имеет два аспекта: программный и аппаратный. С точки зрения сетевого программного обеспечения сеть должна выглядеть для пользователя единым пулом разнообразных ресурсов. Пользователю не важно, какой из серверов предоставляет ему те или иные ресурсы. Это позволяет администратору системы более гибко распределять ресурсы по имеющимся в наличии серверам, упрощает и повышает эффективность контроля и управления ресурсами сети. Для управления ресурсами сети во всех современных ОС выделяются специальные сервисы. В NetWare 4.x/5.x эту роль играют служба каталогов NetWare Directory Services (NDS), системы управления сетями NMS (NetWare Management System) и ManageWise. С точки зрения аппаратных средств эту задачу можно сформулировать так: обеспечение эффективного доступа пула клиентов к пулу серверов. Практическим решением этой задачи является использование при построении кабельной системы так называемых активных элементов. К ним можно отнести локальные мосты, маршрутизаторы, шлюзы, а также устройства, использующие технологию Ethernet Switch и FDDI-концентраторы.

Ограниченность ОС NetWare и ее сетевых приложений

Операционная система NetWare имеет ограниченные возможности по разработке приложений:

- небольшое число средств для разработки NLM-модулей (в NetWare 3.x – Watcom C, в NetWare 4.11 – Watcom C, NetBasic, в NetWare 5.0 – Watcom C, NetBasic и Java);

• в NetWare 3.x нет встроенной защиты ОП между задачами (нитями); в NetWare 4.1 эта защита реализована только между двумя доменами (кольцами) ОП файлового сервера; в NetWare 4.11 домены не используются, здесь введена процедура восстановления системы после аварийных остановок сервера (ABEND); в NetWare 5.x число доменов не ограничено;

• нет средств организации на файловом сервере виртуальных машин;

• многопроцессорную обработку поддерживают только ОС NetWare 5.x, NetWare 4.x SMP и NetWare SFT III, причем возможности последней ОС весьма ограничены;

• не реализована встроенная распределенная файловая система;

• ОС NetWare 3.x/4.x не поддерживает удаленные процедуры.

Для устранения перечисленных недостатков фирма Novell разработала UNIX-подобную ОС UnixWare, которая лишена отмеченных выше ограничений.

NetWare имеет и некоторые другие недостатки:

• для администрирования необходимо иметь рабочую станцию;

• ОС поддерживает персональные компьютеры с Intel-совместимым процессором;

• при инсталляции операционная система NetWare 3.x/4.x не распознает автоматически параметры сетевых адаптеров и периферийных устройств, их следует указывать вручную (NetWare 5.x не имеет указанного недостатка).

7.3. Примеры использования ОС NetWare

Типы серверов, рабочих станций и коммуникационных узлов

Все устройства, подключаемые к сети с ОС NetWare, можно разделить на три функциональные группы: рабочие станции, серверы сети, коммуникационные узлы.

Рабочая станция (Workstation) – персональный компьютер, подключенный к сети, на котором пользователь сети выполняет свою работу. Каждая рабочая станция обрабатывает свои локальные файлы и использует свою ОС, например, Windows 95/98. При этом пользователю доступны все ресурсы сети. Различают три типа рабочих станций (РС): с локальным диском, бездисковая, удаленная.

На РС с диском (жестким или гибким) ОС загружается с этого локального диска. Бездисковая РС не имеет ни жесткого, ни гибкого диска. Для такой станции ее ОС загружается с диска файлового сервера. Это возможно благодаря специальной микросхеме ПЗУ, устанавливаемой на сетевом адаптере бездисковой станции. Удаленная РС – это станция, которая подключается к локальной сети через телекоммуникационные каналы связи (например, с помощью телефонной сети).

Сервер сети (Server) – компьютер, подключенный к сети и выполняющий для пользователей сети определенные услуги, например, хранение данных общего пользования, печать заданий, обработку запроса к СУБД, удаленную обработку заданий и т. д. По выполняемым функциям можно выделить следующие типы серверов.

Файловый сервер (File Server) – компьютер, хранящий данные пользователей сети и обеспечивающий доступ пользователей к этим данным. Как правило, это компьютер с жестким диском большой емкости, со стриммером и т. п. ОС NetWare обеспечивает одновременный доступ пользователей к данным, расположенным на файловом сервере. Файловый сервер выполняет следующие функции: хранение данных; архивирование данных; согласование изменений данных, выполняемых разными пользователями; передачу данных.

Фирма Novell предлагает для файлового сервера ОС NetWare 3.x, 4.x, 5.x.

Сервер баз данных (SQL-Server) – компьютер, выполняющий функции хранения, обработки и управления файлами баз данных (БД). Сервер баз данных выполняет следующие функции: прием и обработку запросов к СУБД, а также пересылку результатов обработки на рабочую станцию; обеспечение секретности данных; согласование изменений данных, выполняемых разными пользователями; взаимодействие с другими серверами баз данных, расположенными в другом месте.

На платформе NetWare функционируют различные серверы БД: System 10/11 (Sybase), Oracle 7.x/8.x (Oracle), SQLBase(Centura), SQL Server (Btrieve Technologies) и т. д. Под управлением NetWare работает и последняя версия пакета Lotus Notes (IBM). Notes также поддерживается ОС Windows NT, UNIX, OS/2. Этот пакет заслужил звание лучшего продукта поддержки коллективных работ. Пользователи Notes могут работать с объектами разных типов: сообщениями, документами, формами. Кроме этого разработаны программы для обмена данными между Notes и внешними БД, использующими язык SQL, а также шлюзы с системами передачи факсов и электронной почты. Notes постепенно становится мощной сетевой информационной средой, стратегической платформой для решения корпоративных задач и обеспечения обслуживания клиентов.

Сервер прикладных программ (Application Server) – компьютер, используемый для решения прикладных программ пользователей. Фирма Novell рекомендует применить для этих целей сервер с ОС UnixWare.

Коммуникационный сервер (Communications Server) – устройство или компьютер, предоставляющий пользователям локальной сети прозрачный доступ к последовательным портам ввода/вывода коммуникационного сервера. С помощью коммуникационного сервера можно создать разделяемый модем, подключив его к одному из портов сервера. Пользователь, подключившись к коммуникационному серверу, может работать с таким модемом так же, как если бы он был подключен непосредственно к рабочей станции. Коммуникационный сервер может быть организован в NetWare на базе пакета NACS, NetWare Connect или Novell Internet Access Server 4.x (см. рис. 7.4).

Сервер доступа (*Access Server*) – это выделенный компьютер, позволяющий выполнять удаленную обработку заданий. Программы, инициируемые с удаленной рабочей станции, выполняются в многозадачной среде этого компьютера (см. рис. 7.5). От удаленной PC принимаются команды, введенные пользо-

вателем с клавиатуры, а возвращаются результаты выполнения задания. В качестве примера сервера доступа можно назвать средство NAS (см. рис.7.5).

Факс-сервер (Fax Server) – устройство или компьютер, выполняющий рассылку и прием факсимильных сообщений для пользователей локальной сети. Факс-серверы могут быть реализованы разными способами:

с помощью пакета, функционирующего, как группа NLM-модулей на файловом сервере NetWare; например, можно назвать продукт Faxserve 2.0 с (Cheyenne Communications);

с помощью пакета, функционирующего на выделенной рабочей станции с одним или несколькими факс-модемами; примером является продукт Net SatisFaxtion (Intel);

с помощью специального устройства, подключенного к сети, например, аппаратно-программного комплекса FaxPress (Castelle).

Сервер резервного копирования данных (Back Up Server) – устройство или компьютер, который решает задачи создания, хранения и восстановления копий данных, расположенных на файловых серверах и рабочих станциях. В качестве такого сервера может использоваться один из файловых серверов сети.

Все перечисленные выше типы серверов (кроме сервера доступа) могут функционировать на одном файловом сервере в виде пакетов программ или утилит NetWare.

Коммуникационный узел – устройство, обеспечивающее взаимодействие отдельных сегментов локальной сети и сетей между собой.

К коммуникационным узлам сети относятся следующие устройства (см. раздел 3.3): повторители; мосты; коммутаторы; маршрутизаторы; шлюзы.

Протяженность сети, расстояние между станциями, в первую очередь, определяется физическими характеристиками передающей среды (коаксиального кабеля, витой пары и т. д.). При передаче данных в любой среде происходит затухание сигнала, что и приводит к ограничению расстояния. Можно значительно расширить сеть, установив специальный усилитель или повторитель сигналов. Такими устройствами являются повторители, мосты и коммутаторы. Часть сети, в которую не входит устройство расширения, принято называть *сегментом* сети.

Повторитель (Repeater) – устройство, позволяющее расширить сеть за счет подключения дополнительных сегментов кабеля. Повторитель, приняв пакет из одного сегмента, передает его во все остальные. При этом происходит как бы «усиление» сигнала. Повторитель выполняет свои функции на физическом уровне, поэтому он зависит от типа сети (ARCNet, Ethernet) и полностью прозрачен для протоколов, используемых в соединяемых сегментах. Повторитель не выполняет развязку присоединенных к нему сегментов, т. е. одновременно поддерживается обмен данными только между двумя станциями одного или разных сегментов.

Мост (Bridge) – устройство, которое так же, как и повторитель, позволяет объединять несколько сегментов. Мост выполняет свои функции на канальном уровне (Data Link), поэтому, как и повторитель, он зависит от типа локальной сети (Token Ring, Ethernet) и полностью прозрачен для протоколов. В отличие

от повторителя мост выполняет развязку присоединенных к нему сегментов, т. е. одновременно поддерживает несколько процессов обмена данными для каждой пары станций разных сегментов. Каждый мост строит внутреннюю таблицу физических адресов подключенных к сети узлов. Процесс ее заполнения заключается в следующем. Каждый кадр (пакет + заголовок кадра), передаваемый по сети, имеет в своем заголовке физические адреса узлов отправления и назначения. Получив на один из своих портов кадр данных, мост работает по следующему алгоритму. На первом шаге он проверяет, занесен ли адрес узла отправителя кадра в его внутреннюю таблицу. Если нет, то мост заносит его в таблицу и связывает с ним номер порта, на который поступил кадр. На втором шаге проверяется, занесен ли во внутреннюю таблицу адрес узла назначения. Если нет, то мост передает принятый кадр во все сети, подключенные ко всем остальным его портам. Если адрес узла назначения найден во внутренней таблице, мост проверяет, подключен ли сегмент узла назначения к тому же самому порту, с которого пришел кадр, или нет. Если да, то мост отфильтровывает кадр, а если нет, то передает его только на тот порт, к которому подключен сегмент сети узла назначения.

При обработке каждого кадра обычный мост сначала принимает кадр, записывает его в буферную память и только после этого передает кадр в требуемый сегмент. Такая технология обработки называется Store-And-Forward (запомнить и передать). При передаче кадров каждый мост вносит задержку, равную времени получения кадра и времени его обработки. При технологии Cut-And-Through (схватить и передать) используются мосты, которые принято называть *коммутаторами* (Switch). Эти устройства, приняв только заголовок кадра, сразу приступают к его обработке и, определив нужный сегмент, ретранслируют (передают) этот кадр. Время задержки при этом существенно меньше, оно равно времени приема заголовка кадра и времени его обработки.

Маршрутизатор или роутер (Router) – устройство, соединяющее сети одного или разного типа, но использующие одну сетевую ОС или один протокол обмена данными. Маршрутизатор анализирует номер сегмента назначения и направляет кадр по оптимальному маршруту. Он выполняет свои функции на сетевом уровне и поэтому зависит от протоколов обмена данными, но не зависит от типа локальной сети. В настоящее время появились маршрутизаторы, позволяющие анализировать (но не преобразовывать) разные протоколы обмена. Более подробные сведения об этих устройствах приведены в разделе 8.2, где обсуждаются протоколы маршрутизации RIP и NLSP.

Шлюз (Gateway) – устройство, позволяющее организовать обмен данными между сетевыми объектами, использующими разные протоколы обмена данными. Шлюз работает на уровнях выше сетевого. Он не зависит от используемой передающей среды, но зависит от используемых протоколов обмена данными. Как правило, шлюз выполняет преобразования между какими-либо протоколами (например, SPX/IPX – TCP/IP, DECnet – SNA и т. д.).

8. АРХИТЕКТУРА СЕТЕВЫХ ОС NETWARE

Глава посвящена принципам построения и функционирования ОС NetWare. Основные сетевые возможности изложены в контексте их реализации протоколами IPX/SPX, NETBIOS, NCP, TLI, протоколом идентификации услуг SAP, протоколами маршрутизации RIP и NLSP. Здесь изложено, как реализуются расширяемость и открытость ОС, какими средствами обеспечивается высокая производительность и надежность. Подробно описаны механизмы защиты информации и диалоговые интерфейсы.

8.1. Принципы построения и функционирования

Принципы организации передачи данных в ОС NetWare

Как уже отмечалось, NetWare относится к классу ОС с выделенным сервером. Поэтому на рабочих станциях (WS) и файловом сервере (FS) используются различные ОС. Взаимосвязь этих ОС осуществляется посредством кадров, которые передаются по шине, соединяющей станции. Рассмотрим процесс передачи данных или команд от рабочей станции к файловому серверу.

С точки зрения организации взаимосвязи с файловым сервером все прикладные программы и утилиты рабочей станции условно можно разделить на два класса: «клиент-файл» и «клиент-сервер». К первому классу («клиент-файл») относятся программы, выполняющие операции (открытие, ввод/вывод, закрытие) с файлами, которые хранятся на файловом сервере. В DOS и Widows при выполнении операций с файлами вырабатывается прерывание 21H, которое перехватывается оболочкой (запросчиком) рабочей станции (рис. 8.1).

Если файл расположен на локальном диске, то оболочка переадресовывает это прерывание ОС РС. Если файл расположен на файловом сервере, то оболочка открывает так называемое гнездо, используемое в дальнейшем для приема пакетов с файлового сервера. Затем запросчик формирует пакет для передачи его на сервер. Одно из полей этого пакета содержит номер гнезда, совпадающий с номером гнезда нити (задачи) сервера, которая будет обрабатывать этот пакет. Далее оболочка с помощью специальной функции протокола

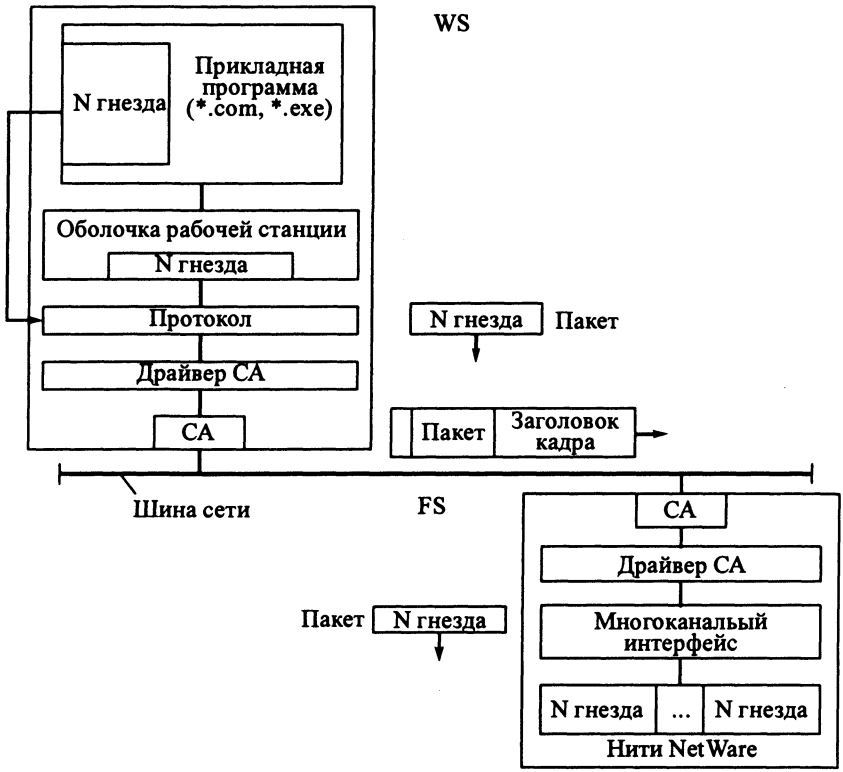


Рис. 8.1. Схема взаимодействия рабочей станции и файлового сервера

направляет пакет в сеть. При прохождении через драйвер сетевого адаптера (СА) пакет превращается в кадр, т. е. к нему добавляется заголовок и концевик кадра. Именно кадр передается по шине сети. Таким образом, оболочка реализует прозрачный доступ прикладной программы к файлу, который хранится на сервере, т. е. программист может использовать обычные функции ввода/вывода языка С или Ассемблера, не вникая в детали API-интерфейса РС с файловым сервером.

При выполнении операций с файлом в программе, функционирующей под управлением OS/2, реализуются аналогичные действия. Только здесь соответствующее прерывание сначала перехватывается ОС, которая переадресовывает его оболочке.

Ко второму классу («клиент-сервер») относятся программы РС, непосредственно взаимодействующие с NLM-модулями файлового сервера. Перед передачей пакета NLM-модулю прикладная программа РС должна открыть гнездо, используемое в дальнейшем для приема пакетов, передаваемых с файлового сервера NLM-модулем. Далее прикладная программа формирует пакет для передачи по сети. Одно из полей этого пакета должно содержать номер гнезда

нити, с которой связан соответствующий NLM-модуль. Затем прикладная программа РС направляет пакет в сеть с помощью специальной функции протокола. При прохождении через драйвер CA пакет превращается в кадр. Таким образом, если программист создал свой NLM-модуль и желает осуществить к нему доступ со стороны рабочей станции, то он должен создать и соответствующую прикладную программу типа «клиент-сервер». При этом он должен знать детали API-интерфейса.

Заголовок кадра, передаваемого по сети, содержит адрес станции получателя и адрес станции отправителя. Если адрес станции получателя совпадает с адресом, на который настроен сетевой адаптер файлового сервера, то кадр принимается и обрабатывается драйвером CA сервера. При этом пакет выделяется из кадра, анализируется соответствующим протоколом и посылается на обработку нити, открывшей гнездо, номер которого совпадает с номером гнезда в пакете. *Нить* – это внутренняя задача NetWare или задача, связанная с каким-либо NLM-модулем. Далее результаты обработки пересылаются прикладной программе, выполняемой на рабочей станции, в виде одного или нескольких пакетов.

Структурная схема ОС

На рис. 8.2 представлена укрупненная структурная схема ОС NetWare. Ядро ОС NetWare загружается в ОП файлового сервера из под DOS (программа SERVER.EXE). Программа SERVER.EXE выполняет следующие действия.

1. Читает из каталога DOS файл STARTUP.NCF и интерпретирует закодированные в нем операторы. Этот небольшой текстовый файл обычно содержит следующие команды:

- оператор загрузки (load) NLM-модуля DOMAIN.NLM (только для версии 4.1); этот модуль читается из каталога DOS и обеспечивает защиту оперативной памяти файлового сервера;
- оператор загрузки NLM-модуля драйвера жесткого диска, например ISADISK.DSK; после этого становится доступной файловая система NetWare. В NetWare 5 не поддерживаются монолитные драйверы *.DSK периферийных устройств, драйверы должны разрабатываться в соответствии

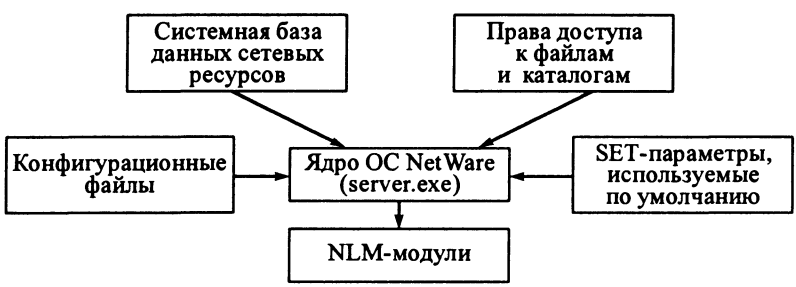


Рис. 8.2. Структурная схема ОС NetWare

с архитектурой периферийного оборудования NetWare (NWPA), согласно которой адаптеры (контроллеры) имеют свои модули (*.HAM), а устройства – свои (*.CDM);

• некоторые SET-команды, например указание максимального уровня вложенности каталогов файловой системы NetWare.

2. Монтирует том SYS файлового сервера и открывает каталог SYSTEM на этом томе.

3. Читает из каталога SYSTEM конфигурационный файл AUTOEXEC.NCF и интерпретирует закодированные в нем операторы. Этот небольшой текстовый файл обычно содержит следующие группы команд:

• некоторые SET-команды, например обеспечивающие переход на летнее и зимнее время;

• SET-команды, определяющие Bindery-контексты в дереве NDS (только для версий 4.x и 5.x);

• операторы, указывающие имя файлового сервера и внутренний номер сети;

• операторы загрузки драйверов сетевых адаптеров (например, NE2000. LAN) и их связи с протоколом IPX и IP;

• операторы загрузки некоторых дополнительных NLM-модулей.

Далее устанавливаются значения SET-параметров, принятые по умолчанию. Изменяя SET-параметры, можно оптимизировать работу ОС. Эти параметры следует изменять с помощью SET-команд, которые можно включать в конфигурационные файлы STARTUP.NCF и AUTOEXEC.NCF или вводить с консоли файлового сервера.

В процессе функционирования ядро выполняет также роль диспетчера нитей (задач) ОС. Каждая нить или связана с каким-либо NLM-модулем, или представляет внутреннюю задачу ОС. NLM-модуль – это исполняемый файл ОС NetWare 3.x и 4.x/5.x.

Системная БД сетевых ресурсов является частью ОС и играет роль надежного хранилища системной информации об объектах, их свойствах (атрибутах) и значениях этих свойств.

ОС NetWare поддерживает описание различных типов объектов: пользователей, групп, файловых серверов, очередей печати, серверов печати и т. д. Каждый из этих типов объектов имеет свой набор свойств. Например, объект «пользователь» характеризуется следующими атрибутами: пароль, балансовый счет, список групп, участником которых является пользователь, и т. д. Значением атрибута (свойства) является та совокупность данных, которая содержится в полях этого атрибута. Системная БД представляет собой множество файлов, хранящихся на томе SYS файлового сервера. В NetWare 3.x и 4.x/5.x эти базы организованы по-разному. В NetWare 3.x она представлена в виде БД Bindery, а в NetWare 4.x/5.x – в виде глобального сетевого каталога NDS. Система каталогов NDS стала мощным средством управления большими корпоративными сетями.

В ОС NetWare данные о защите файлов и каталогов отделены от системной БД и хранятся в элементах DET томов файлового сервера.

Функциональная схема ОС. Модули загрузки NLM

В NetWare 2.x VAP-модули (аналоги NLM-модулей) загружаются в ОП только один раз вместе с ОС. Начиная с NetWare 3.x, NLM-модули могут загружаться в ОП и выгружаться из нее с консоли файлового сервера в динамическом режиме. NLM-модули – представляет собой программы, в которых используется API-интерфейс для связи со службами NetWare. Их разрабатывают, как правило, с помощью компилятора Watcom C, языков NetBasic (начиная с версии 4.11) и Java (начиная с версии 5.0).

Компилятор Watcom C генерирует код, использующий преимущества архитектуры процессоров 80386 и 80486. Он использует 32-битовые ближайшие указатели (near pointers) и 4-байтовые целые числа. Использование 32-битовых указателей делает концепцию моделей памяти во многом ненужной. NLM-модули компилируются для непрерывной модели памяти с абсолютной адресацией («плоской» модели), в которой сегментация памяти гораздо менее важна, чем в других. Одного 32-битового указателя достаточно для адресации всей доступной памяти. Кроме того, компилятор генерирует код, выполняемый в виртуальном режиме.

Все NLM-модули условно можно разделить на две группы: основные (без которых не может функционировать NetWare) и дополнительные. Как видно из рис. 8.3 в качестве NLM-модулей выступают программы с расширениями *.DSK

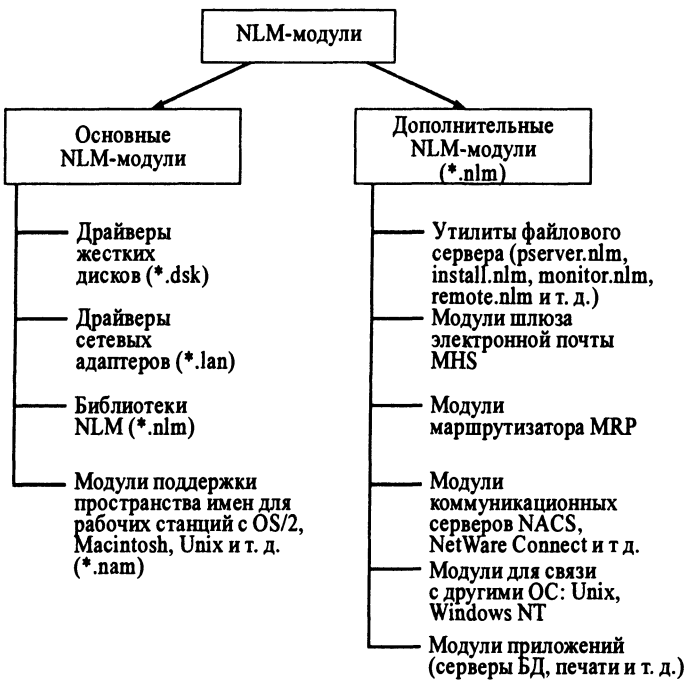


Рис. 8.3. Функциональная схема ОС NetWare

(для версии 5.0 – *.HAM и *.CDM), *.LAN, *.NAM, *.*NLM*. При загрузке NLM-модулей в ОП автоматически создается нить (задача), связанная с этим модулем. Нить может динамически порождать другие нити. Нити выполняются на сервере независимо друг от друга. Синхронизация между ними осуществляется с помощью семафоров.

Диспетчеризация процессов (нитей)

Операционная система NetWare включает в себя следующие очереди (рис. 8.4), в которых находятся различные нити, ожидая освобождения центрального процессора (ЦП): WorkToDoList (только для версии 4.x); RunList; Delayed WorkToDo; LowPriority.

Очереди перечислены в порядке убывания приоритетов обслуживания нитей. Внутри каждой очереди нити диспетчируются в соответствии с дисциплиной FIFO: «первый пришел, первый обслужен». Уже отмечалось, что нить – это или внутренняя задача ОС, или задача, связанная с NLM-модулем. ОС идентифицирует и отслеживает каждую нить по ее блоку управления процессом PCB (Process Control Block).

Обычно в NetWare нить сама себя переводит в неактивное состояние (свтает в очередь). Это происходит в одном из следующих случаев (см. рис. 8.4).

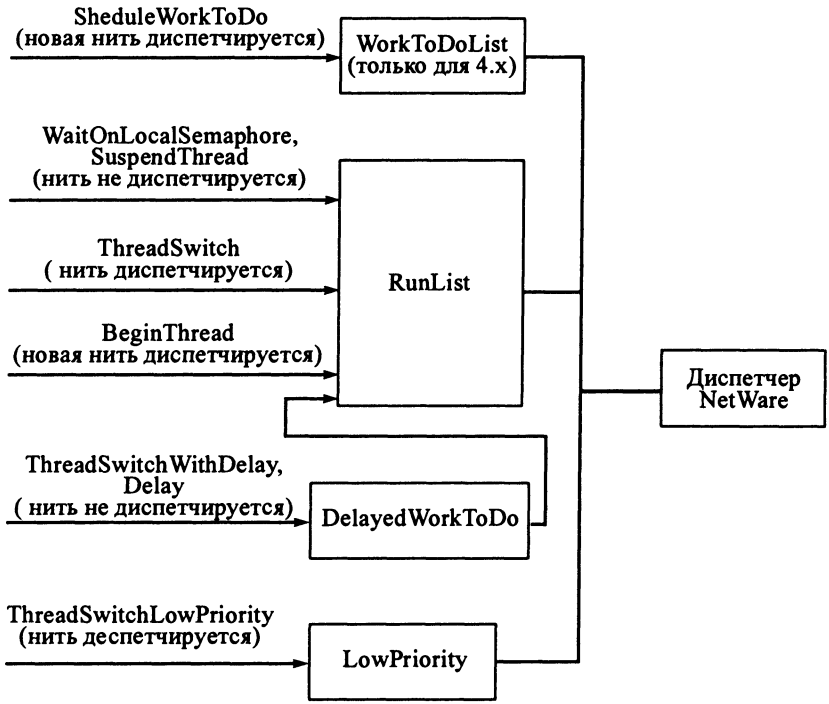


Рис. 8.4. Очереди к процессору

1. Нить выполняет функцию SheduleWorkToDo (для версии 4.x). Создается новая нить, которая заимствуется из ядра NetWare и помещается в очередь WorkToDoList, имеющую высший приоритет для планирования на центральном процессоре (ЦП). Старая нить помещается в конец очереди RunList.

2. Нить приостанавливается, устанавливая семафор (функция WaitOn LocalSemaphore) или ожидая активизации со стороны другой нити (функция SuspendThread). В этом случае нить помещается в конец очереди RunList, но не диспетчируется (не планируется) до наступления требуемого события.

3. Нить выполняет функцию ThreadSwitch, чтобы переключить контекст (т. е. чтобы активизировать другую нить из очереди). В этом случае нить помещается в конец очереди RunList и диспетчируется, когда до нее доходит очередь.

4. Нить выполняет функцию BeginThread. Создается новая нить, которая помещается в конец очереди RunList. Старая нить продолжает выполняться.

5. Нить выполняет функцию ThreadSwitchWithDelay. Нить помещается в конец очереди DelayedWorkToDo и приостанавливается на 50 переключений контекста (нитей), после чего она помещается в конец очереди RunList. Число переключений контекста (50) можно изменить с помощью функции SetThread Handicap (при этом говорят, что устанавливается постоянный гандикап). Часто функцию ThreadSwitchWithDelay используют для того, чтобы активизировать задачи из очереди LowPriority, так как нити из этой очереди выполняются только в том случае, если пуста очередь RunList. Аналогичные действия выполняются, если встречается функция Delay (задержать нить на определенный интервал времени).

6. Нить выполняет функцию ThreadSwitchLowPriority. В этом случае она помещается в очередь LowPriority, имеющую самый низкий приоритет. Нити в этой очереди выполняются только в том случае, если пуста очередь RunList и нет нитей, для которых установлен постоянный приоритет. Типичные низкоприоритетные нити – это создание резервной копии или упаковка файла.

Распределение и защита основной памяти

На рис. 8.5 представлена структура оперативной памяти файлового сервера NetWare 3.x. Структура ОП для NetWare 4.x отличается тем, что области памяти Permanent Memory Pool и Alloc Short Term Memory Pool объединены в один пул Allocated Memory Pool.

В системной области располагается ОС DOS, модуль SERVER.EXE, программы BIOS. В пулах памяти хранятся буферы приема пакетов, таблица соединений, таблица открытых файлов, блоки, динамически выделяемые NLM-модулям, и т. д. Всю оставшуюся память занимает кэш-буфер (Cache Buffers), в котором выделяется кэш неперемещаемой памяти (Cache Non-Movable Memory) и кэш перемещаемой памяти (Cache Movable Memory). В кэше перемещаемой памяти в основном хранятся кэш-таблицы, которые могут быть перемещены ОС в другое место ОП в случае возникновения фрагментации

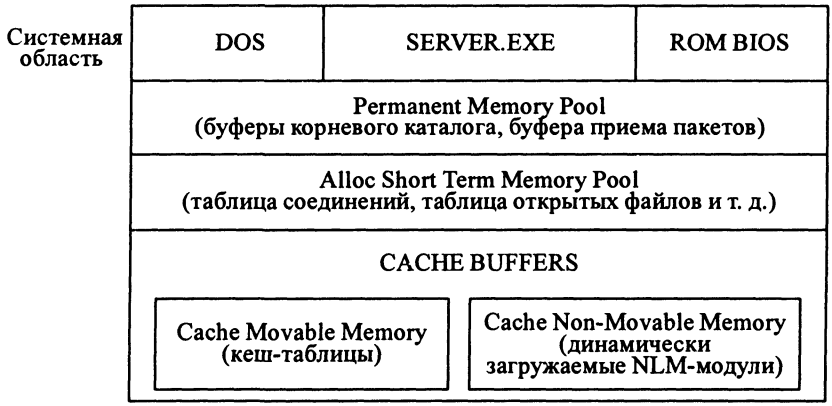


Рис. 8.5. Структура ОП файлового сервера NetWare 3.x

памяти. Кэш неперемещаемой памяти расширяется, если в ОП загружается NLM-модуль. NLM-модули не являются перемещаемыми. После выгрузки NLM-модуля из ОП освободившаяся память вновь возвращается в кэш-буфер.

В NetWare 3.x защита ОП не предусмотрена: любой NLM-модуль имеет доступ к любой области ОП. Поэтому в NetWare 3.x нельзя отлаживать новые NLM-модули на работающей системе, так как ошибка в программе может привести к «зависанию» всей системы.

NetWare 4.1 предусматривает кольцевую и доменную защиту ОП на основе сегментации и страничной организации процессоров Intel 80386/80486. В версии NetWare 4.11 домены не используются, здесь введена процедура восстановления системы после аварийных остановок сервера (ABEND). В версии NetWare 5.x используется доменная защита, но число доменов не ограничено.

Рассмотрим, как процессор выполняет обращение к ОП из программы файлового сервера NetWare 4.1 (рис. 8.6). Адрес ОП состоит из селектора и смещения. 13-битовый индекс селектора определяет дескриптор в таблице страниц. Из этого дескриптора извлекается 32-битовый требуемой страницы в ОП и к нему прибавляется 16-разрядное смещение. В результате образуется требуемый физический адрес данных в ОП. Но перед формированием физического адреса определяется возможность доступа программы к требуемой странице. Для этого в поле RPL селектора адреса копируется 2-битовый уровень привилегий из селектора сегмента кода, загруженного в регистр CS. Таким образом, поле RPL определяет уровень доступа программы к странице. Если значение RPL не превышает значения DPL дескриптора, то программе разрешается доступ к странице. Описанный механизм ограничения доступа называется *кольцевой защитой* памяти. NetWare 4.1 поддерживает только два уровня доступа: 0 и 3 (значения поля RPL). Уровни 1 и 2 эквивалентны уровню 3. В NetWare 4.1 самый привилегированный уровень 0 обозначается как OS, а уровень 3 – OS_PROTECTED.

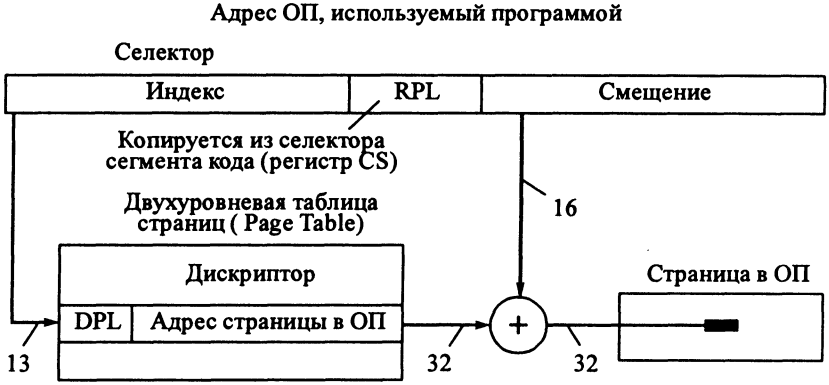


Рис. 8.6. Организация доступа программы к ОП

Доменная защита памяти в NetWare 4.1 заключается в том, что для каждого уровня доступа (OS и OS_PROTECTED) определена своя двухуровневая таблица страниц (Page Table). Работающий в кольце (домене) процесс не может видеть адреса памяти, не отображенные явно на этот домен. Следовательно, для одного домена логически не существует пространства памяти (адресов) другого домена.

Таким образом, в NetWare 4.1 ОП можно разделить на два домена (рис. 8.7). В домен OS загружаются ядро ОС и системные NLM-модули, а в домен OS_PROTECTED можно загружать отлаживаемый NLM-модуль. При попытке несанкционированного доступа к страницам домена OS отлаживаемый NLM-модуль будет аварийно завершен. Таким образом, в NetWare 4.1 можно отлаживать новые программы на работающей системе. После отладки модуля его рекомендуется загружать в домен OS. Это связано с тем, что время переключения между доменами достаточно велико.

Для создания двухдоменной структуры памяти достаточно перед загрузкой NetWare 4.1 поместить в конфигурационный файл команду

LOAD DOMAIN

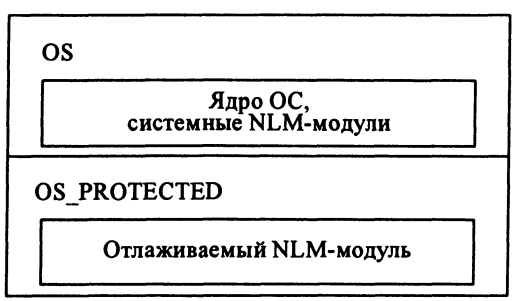


Рис. 8.7. Домены NetWare 4.1

Далее в процессе функционирования системы с консоли файлового сервера можно вводить следующие команды:

DOMAIN = OS

– последующие NLM-модули, загружаемые по LOAD, будут принадлежать домену OS:

DOMAIN = OS_PROTECTED

– последующие NLM-модули, загружаемые по LOAD, будут принадлежать домену OS_PROTECTED.

Структура и управление внешней памятью

Рассмотрим традиционную файловую систему ОС NetWare. На рис. 8.8 представлена логическая структура жесткого диска, установленного на файловом сервере. Один из дисков файлового сервера должен иметь раздел DOS (NetWare загружается из под DOS). Как правило, все остальное пространство диска отводится под раздел NetWare, который делится на тома. В свою очередь, каждый том состоит из сегментов, а каждый сегмент – из блоков. В табл. 8.1 перечислены ограничения, накладываемые на структуру внешней памяти NetWare.

Каждый том NetWare имеет таблицу записей каталога DET (Directory Entry Table) и таблицу размещения файлов FAT (File Allocate Table).

Каждая запись DET соответствует файлу или подкаталогу корневого каталога тома. Эта запись имеет сложную структуру и, в частности, включает имя файла (или подкаталога) и указатель на элемент FAT, соответствующий первому блоку файла (рис. 8.9). Между элементами FAT и блоками тома NetWare

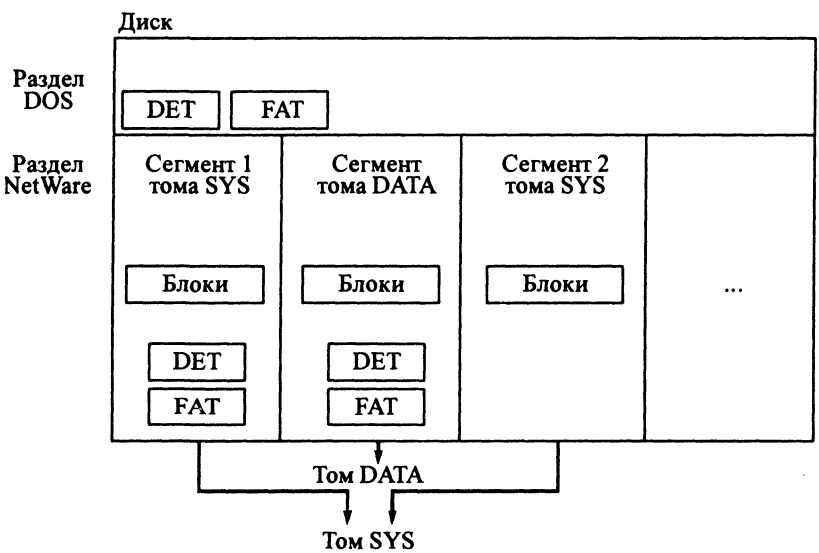


Рис. 8.8. Логическая структура жесткого диска файлового сервера

Таблица 8.1. **Ограничения, накладываемые на традиционную структуру внешней памяти**

Количество жестких дисков на один сервер	Количество томов на один сервер	Количество сегментов на том	Количество сегментов на диск	Размер блока тома (Кб)
До 32	До 64	До 32	До 8	64 (для 4.x/5.x), 32, 16, 8,4

существует взаимно однозначное соответствие. Если размер файла превышает размер блока, то элемент FAT содержит ссылку на другой элемент и т.д. Рассмотренные на рис. 8.9 связи справедливы и для подкаталога. Только здесь каждый блок данных имеет такую же структуру, что и таблица DET.

В таблице DET хранятся следующие типы записей: файлов (File Entries), каталогов (Directory Entries), опекунов (Trustee Entries). Каждая запись имеет длину 128 байт.

1. Запись файла (File Entries) включает следующие поля:
- имя файла,
- идентификатор хозяина файла,
- атрибуты файла,
- размер файла,
- указатель на каталог, где хранится файл,
- дата и время последней модификации,
- имя пространства имен,
- фильтр (маска) наследуемых прав,
- первые шесть опекунских назначений; каждое назначение состоит из 4-байтового идентификатора объекта (ID) и байта прав (Rights), которые имеет этот объект по отношению к файлу; остальные опекунские назначения хранятся в записях опекунов (Trustee Entries),
- указатель на элемент таблицы FAT.

2. Запись каталога (Directory Entries) имеет следующие поля:
- имя каталога,

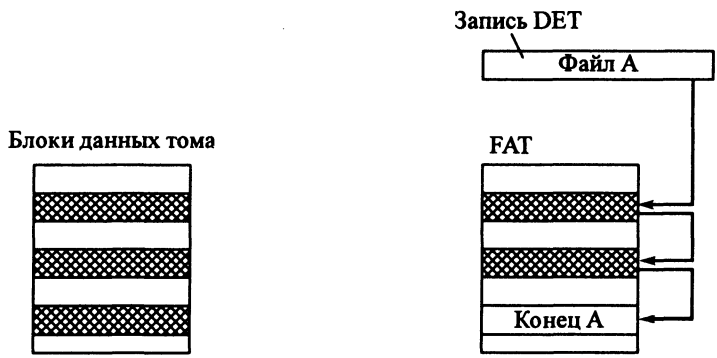


Рис. 8.9. Связь между таблицами DET и FAT

- дата и время создания каталога,
- атрибуты каталога,
- фильтр (маска) наследуемых прав,
- первые шесть опекунских назначений,
- указатель на элемент таблицы FAT.

3. Запись опекунов (Trustee Entries) состоит из следующих полей:

- указатель на запись DET файла или каталога,
- список опекунских назначений (от 2 до 16),
- указатель на следующую запись опекунов.

Опекунские назначения для файлов и каталогов NetWare хранятся в записях DET.

Управление внешней памятью реализуется с помощью утилиты файлового сервера INSTALL.NLM (в версии 5.0 используется утилита NWCONFIG.NLM). Эта диалоговая программа позволяет изменить структуру внешней памяти, а именно: создать новый том файлового сервера, создать новые сегменты существующего тома.

Чтобы в DOS увеличить размер логического раздела, необходимо полностью переинсталлировать жесткий диск. Чтобы в NetWare увеличить размер тома, достаточно просто создать новый сегмент тома на любом диске, где имеется свободное пространство.

В NetWare 4.x/5.x существуют три дополнительные возможности по управлению томом с помощью утилиты INSTALL.NLM (NWCONFIG.NLM).

1. Можно установить флаг File Compression, позволяющий выполнять сжатие файлов тома. При этом возможно автоматическое и ручное сжатие.

Если файл не использовался несколько дней, то он автоматически сжимается. Это число дней устанавливается с помощью SET-параметра Days Untouched Before Compression (категория File System). По умолчанию это значение равно 7. Чтобы этот режим работал не следует выключать сервер на ночь.

Устанавливая атрибут IC с помощью утилиты командной строки FLAG, администратор может вручную выполнить «мгновенное» сжатие файлов. Например, после выполнения команды

FLAG *.* + IC

будут сжаты все файлы в текущем каталоге NetWare.

2. Можно установить флаг Block Suballocation, позволяющий использовать полублоки при размещении файлов тома. Предположим, что размер файла составляет 5 кб, а размер блока тома, где располагается файл, равен 4 кб (рис. 8.10).

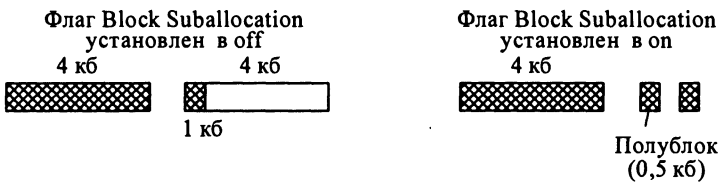


Рис. 8.10. Демонстрация использования полублоков

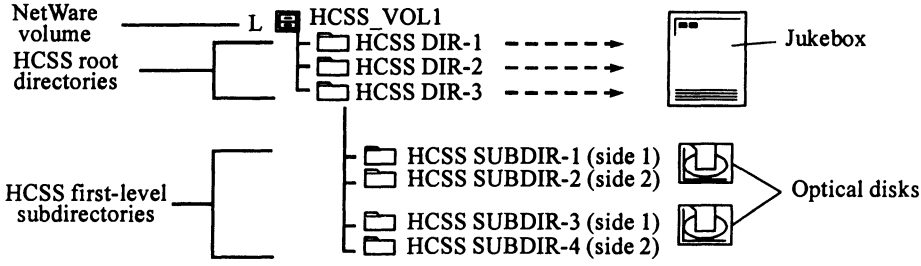


Рис. 8.11. Пример структуры директорий HCSS

Если флаг выключен, то этот файл занимает два блока, причем 3 кб второго блока не используются (то же самое происходит и в NetWare 3.x). Если флаг включен, то данный файл будет занимать один полный блок (4 кб) и два полублока по 0,5 кб. Остальные шесть полублоков (6 × 0,5 = 3 кб) используются другими файлами.

3. Можно установить флаг Data Migration (только на одном томе файлового сервера), позволяющий организовать миграцию данных тома NetWare на магнитооптические диски. Это реализуется с помощью системы поддержки накопителей высокой емкости HCSS (HighCapacity Storage System). Для установки файловой системы HCSS необходимо на PC с помощью утилиты NetWare Administrator выполнить следующие шаги (описание вспомогательных деталей здесь опускается):

1) создать корневые каталоги HCSS;

2) создать подкаталоги HCSS первого уровня. Каждый подкаталог первого уровня ассоциируется с одной стороной оптического диска (рис. 8.11). Подкаталоги 2-го, 3-го и следующих уровней, а также файлы могут быть созданы обычными средствами (например, с помощью Norton);

3) установить верхний и нижний порог емкости для HCSS-тома файлового сервера. Когда при работе с HCSS-томом будет достигнут верхний порог его заполнения, NetWare начинает перемещать файлы с HCSS-тома на оптический диск. Файлы, хранящиеся в подкаталоге первого уровня мигрируют на соответствующую сторону оптического диска. Процесс миграции продолжается до тех пор, пока не будет достигнут нижний порог заполнения HCSS-тома. Перемещение выполняется по принципу LRU (Least Recently Used): миграции подвергаются файлы, к которым дольше всего не было обращения. Даже после выгрузки файла пользователь продолжает видеть его имя в подкаталоге HCSS-тома. При обращении к выгруженному файлу он обратно перемещается с оптического диска в соответствующий подкаталог.

Сетевая файловая система

Одной из основных целей использования сетей является обеспечение доступа всех пользователей к общим устройствам хранения информации, в основном, к жестким дискам. Организация файловой системы во многом схожа с

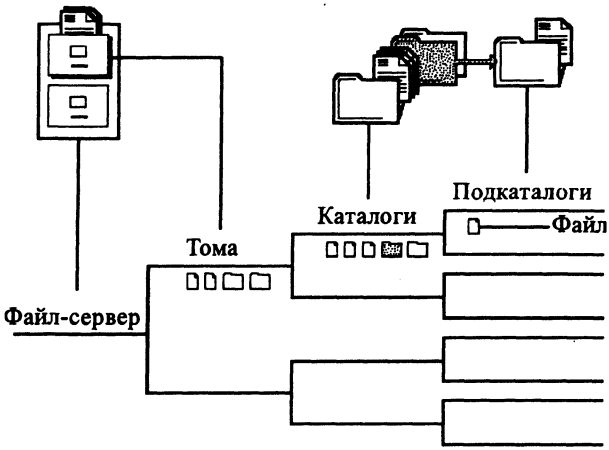


Рис. 8.12. Структура файловой системы

организацией файловой системы DOS, но есть и существенные отличия. Как и в DOS, информация хранится в файлах. Файлы размещаются в древовидной структуре каталогов и подкаталогов. Корнем такого дерева, в отличие от драйва DOS, является том. Тома располагаются на серверах. При наличии соответствующих прав пользователь может получить доступ к томам всех серверов, доступных в сети. Общая структура файловой системы приведена на рис. 8.12.

Рассмотрим элементы этой системы.

Том – это высший уровень файловой системы NetWare. Тома создаются в процессе инсталляции файлового сервера и в процессе его функционирования. В отличие от драйвов DOS, которые соответствуют непрерывным областям на жестком диске, тома могут состоять из нескольких сегментов, которые находятся на одном жестком диске или на разных.

Каталоги – правила работы с каталогами в NetWare и DOS практически совпадают. В отличие от DOS в NetWare ограничивается степень вложенности каталогов (SET-параметр Maximum Subdirectory Tree). По умолчанию в NetWare максимальный уровень вложенности равен 25.

Файлы – правила использования файлов в NetWare такие же, как и в DOS. Файлы могут размещаться в каталогах и подкаталогах тома, включая и корневой.

При инсталляции файлового сервера создается по крайней мере один том с именем SYS. Он предназначен для хранения файлов самой ОС NetWare, а также программ и утилит коллективного пользования. При инсталляции на этом томе создается несколько каталогов (табл. 8.2).

Таблица 8.2. **Системные каталоги ОС NetWare на томе SYS**

Каталог	Описание
LOGIN	Содержит программы, необходимые для подключения к сети
PUBLIC	Содержит основные утилиты NetWare, которые используются клиентами и администратором сети
SYSTEM	Содержит файлы, используемые ОС NetWare или администратором сети. В частности здесь хранятся системные NLM-модули
MAIL	1. Для NetWare 3.x. Используется ОС. Для каждого пользователя в этом каталоге создается отдельный подкаталог с именем, совпадающим с шестнадцатеричным идентификатором (ID) этого пользователя из БД Bindery. В этом подкаталоге, в частности, хранится пользовательская процедура подключения (login script). 2. Для NetWare 4.x/5.x. В основном, данный каталог предназначен для различных почтовых систем, совместимых с NetWare 4.x/5.x. Личные подкаталоги в этом каталоге создаются только: для клиента ADMIN при инсталляции для обеспечения воз-можности работы в режиме эмуляции Bindery, для клиентов, создаваемых автоматически при выполнении Upgrade с версии 3.x; при этом личные процедуры регистрации перемещаются в дерево NDS в качестве свойства объекта USER. Если пользователь описывается обычным способом с помощью средств NDS, то подкаталог не создается
ETC	Содержит файлы примеров, помогающих конфигурировать сервер для работы с протоколом TCP/IP
DELETED.SAV	Каталог с этим именем находится в корне каждого тома. Если вместе с файлами был удален и сам каталог, то эти файлы перенаправляются в каталог DELETED.SAV и их следует восстанавливать, в случае необходимости, в этом каталоге
DOC	В этом каталоге инсталлируется документация в электронном виде
DOCVIEW	Содержит средства просмотра электронной документации
QUEUES (только для NetWare 4.x/5.x)	Содержит очереди на печать

Войдя в сеть, можно создавать другие каталоги. Пользователи могут обмениваться файлами через эти каталоги и хранить в них свои собственные файлы. Однако, прежде чем использовать созданные каталоги, необходимо, во-первых, описать пользователей в системе и, во-вторых, наделить их правами, необходимыми для доступа к каталогам.

Пользователь осуществляет доступ к файлам и каталогам NetWare с рабочей станции, на которой установлена своя ОС, например DOS. Связывание драйвов DOS с томами NetWare выполняется с помощью утилиты командной строки MAP. Например, после выполнения команды

MAP F: = FS4S/SYS:

том SYS файлового сервера FS4S планируется на драйв F: и становится доступным операционной системе DOS. Такие драйвы называют *логическими устройствами.*

Известно, что в DOS пути поиска указываются с помощью параметра окружения PATH. Чтобы указать ОС DOS пути поиска на файловом сервере, следует также использовать команду MAP, но с другими параметрами. Например, после выполнения команды

MAP S1: = FS4S/SYS:PUBLIC

будет создан драйв Z: (выбираются буквы с конца латинского алфавита), спланированный на каталог PUBLIC тома SYS файлового сервера FS4S. При этом путь Z: будет добавлен в начало параметра PATH. Создаваемые по MAP драйвы Z:, Y: и т. д. называются *поисковыми устройствами.*

Сетевая печать

Все клиенты сети могут пользоваться одним или несколькими общими принтерами. На рис. 8.13 представлена схема организации сетевой печати в NetWare. При использовании сетевой печати данные, направляемые на печать, помещаются в *очередь* в виде задания. *Сервер печати* периодически сканирует очереди и при наличии в них заданий на печать пересылает их на *принтеры*. Рассмотрим элементы сетевой печати.

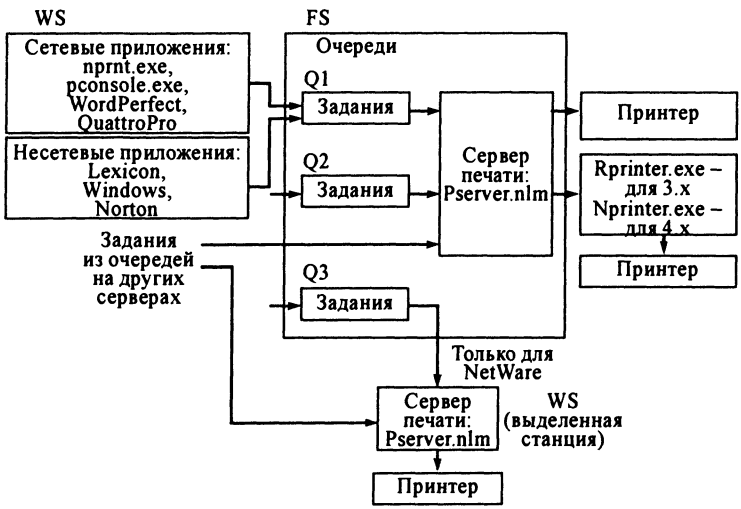


Рис. 8.13. Схема организации сетевой печати

Очереди. Когда PC посылает данные на печать, то они временно сохранятся в виде файла в специальном каталоге. Этот файл называется заданием, а специальный каталог – очередью. В NetWare 3.x очередь представляет собой подкаталог каталога SYSTEM тома SYS. Имя подкаталога имеет расширение QDR, например SYS:SYSTEM\09000001.QDR. В этом подкаталоге находятся файлы, определяющие параметры очереди (*.SRV, *.SYS), и файлы с заданиями на печать (*.Q). В файле с расширением SRV имеется ссылка на серверы печати, обслуживающие данную очередь. В файле с расширением SYS хранится информация, необходимая серверу печати для поддержки очереди: номер станции, передавшей задание, идентификационный номер пользователя, имя файла задания на печать, время постановки его в очередь, заданное время начала печати и т. д. В каждой очереди имеется по одному файлу с расширением SRV и SYS. При добавлении новых заданий информация в этих файлах обновляется. Задание на печать хранится в виде файла с расширением Q. При формировании имени этого файла используются идентификационный номер очереди и порядковый номер задания в ней, например, 00090001.Q, 00090002.Q и т. д.

В NetWare 4.x/5.x подкаталоги очередей могут быть расположены на любом томе файлового сервера (в версии NetWare 3.x подкаталоги очередей всегда создаются в каталоге SYSTEM тома SYS). Если на томе сохраняется хотя бы одна очередь, то в его корне автоматически создается каталог QUEUES, подкаталоги которого и являются очередями. Параметры очередей хранятся в дереве NDS как свойства объекта Print Queue.

Сервер печати. Сервер печати – программа, которая постоянно сканирует очереди на печать и направляет задания из очередей на принтеры. В NetWare 3.x сервер печати может выполняться либо в виде NLM-модуля на файловом сервере, либо в виде EXE-файла на выделенной рабочей станции (см. рис. 8.13). Для каждого сервера печати создается подкаталог в каталоге SYS:SYSTEM. Его имя совпадает с 16-ричным идентификатором соответствующего объекта Print Server из базы данных Bindery. Он содержит файлы со служебной информацией, требуемой для работы самого сервера. В файле FILESERV размещены данные об обслуживаемых файловых серверах. В этом же подкаталоге находятся файлы с именами PRINT.* (информация для каждого определенного принтера), QUEUE.* (сведения об очередях вывода на печать) и NOTIFY.* (списки пользователей, которых нужно уведомлять при возникновении проблем с принтером). Информация, содержащаяся в файлах с одинаковым расширением (например, PRINT.000, QUEUE.000 и NOTIFY.000), относится к одному принтеру.

В NetWare 4.x/5.x сервер печати реализован в виде NLM-модуля, т. е. может быть загружен только на файловом сервере. На одном файловом сервере может быть загружен только один сервер печати (это справедливо и для NetWare 3.x). Параметры сервера печати хранятся в дереве NDS как свойства объекта Print Server.

Принтеры. Принтеры в сетях NetWare можно подключать тремя способами.

1. К файловому серверу. К файловому серверу можно подключить пять принтеров (к трем параллельным и двум последовательным портам). Каждый сервер печати может обслуживать в NetWare 3.x до 16 принтеров, в NetWare 4.x/5.x – до 256 принтеров.

2. К любой PC, функционирующей под управлением DOS или OS/2. В этом случае PC можно использовать в обычном режиме. На этой станции требуется вручную загружать необходимое программное обеспечение сетевого принтера (RPRINTER.EXE – для NetWare 3.x, NPRINTER.EXE – для NetWare 4.x/5.x).

3. Непосредственно к сетевой шине, если принтер снабжен специальной сетевой платой.

В NetWare 4.x/5.x параметры принтера хранятся в дереве NDS как свойства объекта Printer.

Для организации сетевой печати необходимо выполнить следующие действия.

1. При необходимости описать с помощью утилиты PRINTDEF.EXE (для 3.x) или NetWare Administrator (для 4.x и 5.x) новые режимы печати (ESC-последовательности, которые должны быть выполнены перед началом печати), новые формы печати (количество строк на странице и число символов в строке).

2. Описать с помощью утилиты PCONSOLE.EXE (для 3.x) или NetWare Administrator (для 4.x и 5.x) объекты очередей, серверов печати, принтеров.

3. Описать с помощью утилиты PRINTCON.EXE (для 3.x) или NetWare Administrator (для 4.x и 5.x) конфигурации заданий на печать (Print Job Configuration): заголовок печати, число копий, очередь по умолчанию и т.д.

4. Запустить сервер печати на файловом сервере (PSERVER.NLM) или на выделенной рабочей станции (PSERVER.EXE – только для NetWare 3.x).

Ниже приведен пример организации печати из-под WINDOWS.

CAPTURE J=J1	Начать перехват
WIN	Запустить WINDOWS
Печать из приложения WINDOWS (например, из WinWord)	Данные, направляемые в LPT-порт, перехватываются и передаются в сетевую очередь, а затем распечатываются на сетевом принтере. Параметры печати выбираются из описания конфигурации задания на печать J1
Выгрузить WINDOWS	
CAPTURE EC	Завершить перехват
WIN	Вновь запустить WINDOWS
Печать из приложения WINDOWS	Данные распечатываются на принтере, который подключен к локальному LPT-порту
Выгрузить WINDOWS	

5. При необходимости с помощью утилиты PCONSOLE.EXE (для 3.x) или NetWare Administrator (для 4.x и 5.x) выполнить управление печатью (изменить приоритет очереди и местонахождение задания в очереди, задержать задание в очереди, запретить клиенту направлять задание в очередь, запретить серверу печати обслуживать очередь и т. д.).

Печать в сети осуществляется: из сетевых приложений. Так называют приложения, в которых используется API-интерфейс службы сетевой печати. В качестве примера можно назвать утилиту NPRINT.EXE и пакеты WordPerfect, QuattroPro; из несетевых приложений. Так называют приложения, в которых данные, выводимые на печать, направляются в LPT-порт PC. Чтобы перехватить эти данные и передать их в сетевую очередь, используют утилиту командной строки CAPTURE. В качестве примера несетевых приложений можно назвать WINDOWS, LEXICON, NORTON.

8.2. Основные сетевые возможности

Протокольный набор IPX/SPX

Выше отмечалось, что взаимодействие между станциями осуществляется с помощью кадров. Пакет является частью кадра и имеет свой заголовок. В дальнейшем под протоколом будем понимать системную программу, которая обрабатывает определенные поля кадра.

NetWare поддерживает следующие уровни протоколов в классификации OSI:

- канальный, обрабатывающий заголовок кадра (драйвер сетевого адаптера);
- сетевой (IPX, SPX, IP, NETBIOS, TLI);
- транспортный (SPX, NCP, NETBIOS, TLI);
- сессионный (NETBIOS, NCP);
- прикладной (RIP, NLSP, SAP и др.).

Для рабочей станции с ОС DOS протоколы IPX и SPX входят в состав программы IPXODI.COM, которая загружается с помощью bat-файла STARTNET.BAT.

Протокол IPX (Internetwork Packet Exchange) обрабатывает так называемый пакет IPX, являющийся основным средством, которое используется при передаче данных в сетях NetWare. Формат пакета IPX представлен на рис 8.14.

Все поля, указанные на рис 8.14, кроме последнего (Data), образуют заголовок пакета. Особенностью формата пакета является то, что все поля заголовка содержат значения в перевернутом формате: по младшему адресу записывается старший байт данных.

Рассмотрим подробнее назначение отдельных полей пакета.

Поле Checksum предназначено для хранения контрольной суммы пакета или другой служебной информации. В прикладных программах обычно не используется.

Поле Length определяет общий размер пакета вместе с заголовком. NetWare поддерживает следующие максимальные длины пакетов: Token Ring и ARCnet – 4202 байт, Ethernet – 1514 байт. Это поле устанавливается протоколом IPX передающей станции.

2	Checksum – контрольная сумма
2	Length – общая длина пакета
1	TransportControl – счетчик пройденных маршрутизаторов
1	PacketType – тип пакета
4	DestNetwork – номер сети получателя пакета
6	DestNode – адрес станции-получателя
2	DestSocket – гнездо программы-получателя
4	SourceNetwork – номер сети отправителя пакета
6	SourceNode – адрес станции-отправителя
2	SourceSocket – гнездо программы-отправителя
Длина	Data – передаваемые данные

Рис. 8.14. Структура пакета IPX

Поле TransportControl служит как бы счетчиком маршрутизаторов, которые проходит пакет на своем пути от передающей станции к принимающей. В начале это поле устанавливается в 0 протоколом IPX передающей станции.

Поле PacketType определяет тип передаваемого пакета. Программа, которая передает пакет средствами IPX, должна устанавливать в это поле значение 0x04.

Поле DestNetwork определяет номер сети, в которую передается пакет. Это поле устанавливается в прикладной программе. Если в поле указывается нулевое значение, то пакет предается в сеть (сегмент), к которой подключена станция.

Поле DestNode определяет адрес станции, которой предназначен пакет. Это поле устанавливается прикладной программой. Если пакет предназначен всем станциям в сети (сегменте), то в поле указывается значение FFFFFFFFh.

Поле DestSocket предназначено для определения программы, которая запущена на станции-получателе и должна принять пакет. Это поле устанавливается в прикладной программе.

Поля SourceNetwork, SourceNode, SourceSocket содержат соответственно номер сети, из которой посылается пакет, адрес передающей станции и гнездо программы, которая передает пакет. Эти поля заполняются протоколом IPX передающей станции.

Поле Data в пакете IPX содержит передаваемые данные. Это поле формируется протоколом IPX передающей станции на основании описания блока ECB (рис. 8.15). Блок ECB состоит из фиксированной части размером 36 байт и массива дескрипторов, описывающих отдельные фрагменты передаваемого или принимаемого пакета данных.

Рассмотрим назначение отдельных полей блока ECB.

Поле Link предназначено для организации списков, состоящих из блоков ECB. Устанавливается протоколом IPX.

Поле ESRAddress содержит адрес программного модуля, который получает управление при завершении процесса чтения или передачи пакета IPX. При необходимости устанавливается прикладной программой.

4	Link – указатель на следующий ECB
4	ESRAddress – адрес программы ESR
1	InUse – флаг состояния ECB
1	CCode – код завершения запроса
2	Socket – номер гнезда для приема или передачи
4	IPXWorkspace – рабочий буфер для IPX
12	DriverWorkspace – рабочий буфер
6	ImmAddress – адрес той станции сегмента, которой непосредственно передается пакет
2	FragmentCnt – количество фрагментов в пакете. Каждая следующая пара полей образует дискриптор фрагмента
4	Adress – адрес 1-го фрагмента
2	Size – размер 1-го фрагмента
4	Adress – адрес 2-го фрагмента
2	Size – размер 2-го фрагмента
...	и т. д.

Рис 8.15. Формат блока ECB, используемого в функциях API-интерфейса

Поле InUse служит индикатором завершения операции приема или передачи пакета.

Поле Ccode содержит код результата выполнения функции API-интерфейса.

Поле Socket содержит номер гнезда. Если ECB используется для приема, то в этом поле должен указываться номер гнезда принимающей программы. Если ECB используется для передачи, то поле содержит номер гнезда передающей программы. Заполняется в прикладной программе и используется протоколом IPX для заполнения поля SourceSocket пакета IPX (см. рис. 8.14).

Поля IPXWorkspace и DriverWorkspace зарезервированы для использования протоколом IPX.

Поле ImmAddress содержит при передаче адрес узла сегмента, куда непосредственно будет направлен пакет. Если пакет передается в пределах одного сегмента, поле содержит адрес станции-получателя (такой же, как и в поле DestNode заголовка пакета IPX). Если пакет предназначен для другого сегмента и будет проходить через маршрутизатор, поле ImmAddress содержит адрес этого маршрутизатора. Если пакет предназначен всем узлам сегмента, то в поле указывается значение FFFFFFFFh. При передаче пакета это поле заполняется в прикладной программе. Значение этого поля используется драйвером сетевого адаптера для формирования адреса-получателя в заголовке кадра.

При приеме поле ImmAddress содержит адрес станции сегмента, от которой пришел пакет. В этом случае поле заполняется протоколом IPX. Этот адрес станции выбирается из заголовка кадра (поле «адрес отправителя») и, как правило, используется прикладной программой для передачи ответа.

Поле FragmentCnt устанавливается прикладной программой и содержит число фрагментов, на которое надо разбить принятый пакет, или из которых надо собрать передаваемый пакет, т. е. в программе можно указать отдельные буферы для приема/передачи заголовка и данных пакета. В этом случае значение поля FragmentCnt должно быть равно двум.

Сразу вслед за полем FragmentCnt располагаются дескрипторы фрагментов, каждый из которых состоит из адреса фрагмента (поле Address) и размера фрагмента (поле Size).

Фирма Novell предлагает API-интерфейсы для работы по протоколу IPX на рабочей станции и файловом сервере. В табл. 8.3 перечислены примитивы (функции) этих интерфейсов.

Таблица 8.3. **Примитивы API-интерфейсов для работы протоколов IPX**

Примитив	Описание
IPXOpenSocket	Открыть гнездо. Вход – тип гнезда, номер открываемого гнезда
IPXCloseSocket	Закрыть гнездо. Вход – номер закрываемого гнезда
IPXGetLocalTarget	Применяется для вычисления значения непосредственного адреса. Вход – адрес 12-байтного поля с полным адресом конечной станции-назначения (номер сети – 4 байт, адрес станции – 6 байт, номер гнезда – 2 байт). Выход – значение адреса той станции сегмента (например маршрутизатора или файлового сервера), которой непосредственно передается пакет (значение поля ImmAddress блока ECB)
IPXGetInternetworkAddress	С помощью этого примитива программа может узнать сетевой адрес станции, на которой она работает. Выход – 10-байтовый адрес станции (номер сети – 4 байт, адрес станции – 6 байт)
IPXListenForPacket (для программы на рабочей станции) IPXReceive (для NLM-модуля)	Используется для приема пакета из сети. Вход – адрес блока *ECB*
IPXSendPacket (для программы на рабочей станции) IPXSend (для NLM-модуля)	Используется для передачи пакета в сеть. Вход – адрес блока *ECB*
IPXRelinquishControl (для программы на рабочей станции) ThreadSwitch (для NLM-модуля)	Используется, чтобы освободить процессор для выделения процессорного времени протоколу IPX. Применяется в том случае, если поле InUse блока ECB (см. рис. 8.15) в цикле опрашивается прикладной программой

Как видно из табл. 8.3, имена многих примитивов совпадают для программ на рабочей станции и для NLM-модулей файлового сервера.

Для поддержки на файловом сервере служб протокола IPX используется библиотека CLIB.NLM. Для организации доступа к этим службам необходимо дополнительно загрузить NLM-модуль IPXS.NLM в стек протоколов, основанных на STREAMS.

	Поля IPX
2	Checksum – контрольная сумма
2	Length – общая длина пакета
1	TransportControl – счетчик пройденных маршрутизаторов
1	PacketType – тип пакета
4	DestNetwork – номер сети получателя пакета
6	DestNode – адрес станции-получателя
2	DestSocket – гнездо программы-получателя
4	SourceNetwork – номер сети отправителя пакета
6	SourceNode – адрес станции-отправителя
2	SourceSocket – гнездо программы-отправителя
	Поля SPX
1	ConnControl – управление потоком данных
1	DataStreamType – тип данных в пакете
2	SourceConnID – идентификатор канала отправителя
2	DestConnID – идентификатор канала получателя
2	SeqNumber – счетчик переданных пакетов
2	AckNumber – номер следующего пакета
2	AllocNumber – количество буферов для приема
Длина	Data – передаваемые данные

Рис. 8.16. Структура пакета SPX

Протокол IPX определяет самый быстрый уровень передачи данных в сетях NetWare. Он относится к классу датаграммных протоколов типа «точка-точка» без установления соединения. Это означает, что вашей прикладной программе не нужно устанавливать специальное соединение между ней и получателем. Протокол IPX имеет несколько недостатков:

не гарантирует доставку данных,

не гарантирует сохранение правильной последовательности приема пакетов,

не подавляет прием дублированных пакетов.

Обработка ошибок, возникающих при передаче пакетов IPX, возлагается на прикладную программу, принимающую пакеты. Указанных недостатков не имеет протокол SPX (Sequenced Packet eXchange), ориентированный на установление соединения. Протокол SPX обрабатывает пакет SPX, формат которого представлен на рис. 8.16.

Первые десять полей пакета SPX совпадают с заголовком пакета IPX. Рассмотрим остальные поля заголовка SPX.

Поле ConnControl содержит набор битовых флагов, управляющих передачей данных по каналу SPX.

Поле DataStreamType также представляет собой набор однобитовых флагов, которые используются для классификации данных, передаваемых или принимаемых при помощи протокола SPX.

Поле SourceConnID содержит номер канала связи передающей программы, присвоенный протоколом SPX при создании канала связи. Полем управляет протокол SPX.

Поле DestConnID содержит номер канала связи принимающей стороны. Так как все пакеты, приходящие на один номер гнезда, могут принадлежать разным каналам связи (на одном гнезде можно открыть несколько каналов связи), то приходящие пакеты следует классифицировать по номеру канала связи. Полем управляет протокол SPX.

Поле SeqNumber содержит счетчик пакетов, переданных по каналу в одном направлении. На каждой стороне канала используется свой счетчик. После достижения значения FFFFh счетчик сбрасывается в 0, после чего процесс счета продолжается. Содержимым поля управляет протокол SPX.

Поле AckNumber содержит номер следующего пакета, который должен быть принят протоколом SPX. Содержимым этого поля управляет протокол SPX.

Поле AllocNumber содержит количество буферов, распределенных программой для приема пакетов. Содержимым этого поля управляет протокол SPX.

Для протокола SPX используется точно такой же блок ECB, что и для протокола IPX. Прикладная программа, в которой используются примитивы API-интерфейса с протоколом SPX, обычно включает выполнение следующих шагов:

открыть гнездо для IPX,

установить с помощью функций IPXListenForPacket (или IPXReceive) и IPXSendPacket (или IPXSend) связь между программами, которые должны организовать обмен данными между собой,

с помощью функции IPXOpenSocket открыть гнездо для SPX,

установить канал связи между программами (табл. 8.4),

выполнить обмен данными по установленному каналу связи,

закрыть канал связи.

Фирма Novell предлагает API-интерфейсы для работы по протоколу SPX на PC и файловом сервере. В табл. 8.4 перечислены основные примитивы этих интерфейсов.

Таблица 8.4. **Примитивы API-интерфейсов для работы по протоколу SPX**

Примитив	Описание
SPXListenForConnection	Используется в паре с функцией, SPXEstablish Connection для образования канала связи. Вход – адрес блока ECB. Прикладную программу, в которой используется функция SPXListenForConnection, принято называть программой-сервером (она принимает первый пакет канала). Прикладную программу, в которой используется функция SPXEstablishConnection, называют программой-клиентом (она посылает первый пакет программе-серверу). При использовании примитива SPXListenForConnection в программе-сервере необходимо выполнить следующие шаги: 1. Выполнить обращение к функции SPXListenFor SequencedPacket, чтобы обеспечить в дальнейшем прием пакета от программы-клиента.

Окончание табл. 8.4

Примитив	Описание
SPXListenForConnection	2. Выполнить обращение к функции SPXListenFor Connection. После успешного образования канала в поля InUse и Ccode блока ECB будет записано нулевое значение. 3. Ожидать приема пакета (см. шаг 1)
SPXEstablishConnection	Используется в программе-клиенте для образования канала связи. Вход – адрес блока ECB. При использовании этого примитива в программе-клиенте необходимо выполнить следующие шаги: 1. Выполнить обращение к функции SPXListenFor SequencedPacket, чтобы обеспечить в дальнейшем прием пакета от программы-сервера. 2. Выполнить обращение к функции SPXEstablish Connection. После успешного образования канала в поля InUse и Ccode блока ECB будет записано нулевое значение, а поле SourceConnID заголовка пакета SPX будет содержать номер образованного канала связи. 3. С помощью функции SPXSendSequencedPacket выполнить передачу первого пакета программе-серверу
SPXListenForSequenced Packet	Обеспечивает прием пакета средствами протокола SPX. Вход – адрес блока ECB
SPXSendSequencedPacket	Обеспечивает передачу пакета. Вход – адрес блока ECB, а также номер канала, используемый программой-получателем. Номер канала следует выбирать из поля SourceConnID заголовка принятого пакета
SPXGetConnectionStatus	Проверить состояние канала. Вход – номер канала и указатель на буфер, куда записывается информация о состоянии канала
SPXTerminateConnection	Эта функция автоматически посылает удаленному партнеру пакет, который состоит из одного заголовка. В поле DataStreamType этого заголовка находится значение FEh, которое говорит программе получателя закрыть канал (т. е. выполнить в ответ функцию SPX TerminateConnection). Вход – адрес блока ECB и номер канала связи. Далее обе программы должны закрыть используемые гнезда.
SPXAbortConnection	Функция разрывает канал связи без согласования с программой получателя. Используется только в катастрофических случаях, когда невозможно выполнить нормальную процедуру закрытия канала. Вход – номер канала связи

Как видно из табл. 8.4, имена примитивов совпадают для программ на PC и для NLM-модулей файлового сервера.

Для поддержки на файловом сервере служб протокола SPX используется библиотека CLIB.NLM. Для организации доступа к этим службам следует дополнительно загрузить NLM-модуль SPXS.NLM в стек протоколов, основанных на STREAMS.

Сравнивая рассмотренные выше протоколы IPX и SPX, можно сказать, что протокол IPX быстр, но SPX более надежен.

Протокол NETBIOS

В ОС NetWare протокол NETBIOS является надстройкой над протоколом IPX и используется для организации обмена данными между PC. Протокол NETBIOS реализован в виде резидентной программы NETBIOS.EXE, входящей в комплект поставки NetWare. Для обмена данными между этими резидентными программами используются пакеты IPX с номером гнезда 0x0455 и типом пакета 20.

Для идентификации PC протоколы IPX и SPX используют номер сети, адрес станции в сети и номер гнезда. Адрес станции определяется на аппаратном уровне и представляет собой число длиной 6 байт. Номер сети занимает 4 б. Номер гнезда выделяется динамически протоколом IPX или может быть получен в фирме Novell. Номер гнезда занимает 2 б.

Протокол NETBIOS использует другой механизм адресации станций и программ. Для адресации станций используются имена размером 16 б. Каждая станция имеет одно постоянное имя (permanent name), которое образуется из аппаратного адреса добавлением к нему слева десяти нулевых байтов. Кроме постоянного имени протокол NETBIOS позволяет добавлять (и удалять) обычные имена и групповые имена. Обычные имена служат для идентификации PC, групповые имена могут служить для посылки пакетов одновременно нескольким станциям в сети. Постоянное имя удалять нельзя, так как оно полностью определяется аппаратным обеспечением станции.

При добавлении обычного имени протокол NETBIOS опрашивает всю сеть для проверки уникальности имени. Групповое имя может быть одинаковым на нескольких станциях, поэтому при добавлении группового имени опрос сети не выполняется. После добавления нового имени этому имени присваивается так называемый номер имени (name number), который используется для передачи данных по сети.

Сравнивая методы адресации, используемые протоколами IPX/SPX и NETBIOS, можно заметить, что метод адресации протокола NETBIOS более удобен. Вы можете адресовать данные не только одной станции (как в IPX и SPX) или всем станциям сразу (как в IPX), но и группе станций, имеющим одинаковое групповое имя.

1	Cmd – код команды
1	CCode – код ошибки до выполнения команды
1	LocalSessionNumber – номер канала
1	NetworkNameNumber – номер имени
4	Buffer – дальний указатель на буфер данных
2	Size – длина буфера
16	CallName – имя станции-получателя
16	OurName – имя станции-отправителя
1	ReceiveTimeoute – время ожидания завершения приема пакета
1	SendTimeout – время ожидания завершения передачи пакета
4	PostRoutine – адрес POST-программы
1	AdapterNumber – номер сетевого адаптера
1	FinalCCode – код ошибки после выполнения команды
14	Reserved – рабочий буфер протокола NETBIOS

Рис. 8.17. Формат блока NCB

Чтобы выполнить функцию NETBIOS, в прикладной программе необходимо
- заполнить поля блока NCB (Network Control Block);
- загрузить в регистр ES:BX дальний адрес блока NCB;
- вызвать программное прерывание 5C (INT 5C).

Формат блока NCB представлен на рис. 8.17. Рассмотрим назначение полей блока NCB.

Поле Cmd содержит код команды, которую необходимо выполнить.

Поле CCode содержит код ошибки, возвращаемый после проверки параметров до выполнения команды.

Поле LocalSessionNumber содержит номер канала, установленного с другой программой. Оно используется только при выдаче команд передачи данных через каналы.

Поле NetworkNameNumber содержит номер имени, который присваивается при добавлении обычного или группового имени. Это поле должно быть заполнено при приеме дейтаграмм.

Поле Buffer представляет собой дальний указатель на буфер, который должен содержать данные перед выполнением передачи, или на буфер, который будет использован для приема данных.

Поле Size определяет размер буфера, используемого для приема или передачи данных.

В поле CallName указывается имя станции-получателя.

Поле OurName содержит имя станции-отправителя. Обычно используется в командах создания имени станции или создания канала.

Поля ReceiveTimeout и SendTimeout содержат интервал времени (измеряемый в 1/2 с), в течение которого ожидается завершение соответственно команд приема и передачи.

Поле PostRoutine содержит указатель на программу (POST-программу), которая получает управление после завершения команды.

Поле AdapterNumber используется, если на рабочей станции установлено несколько сетевых адаптеров (в сетях Ethernet этого обычно не бывает). В этом поле указывается номер адаптера, для которого предназначена команда. Первый адаптер имеет номер 0, второй –1 и т. д.

Поле FinalCCode содержит во время выполнения команды значение 0xFF. После завершения выполнения команды в это поле записывается ноль или код ошибки, который относится к выполнению команды в целом (в отличие от кода в поле CCode).

Поле Reserved зарезервировано для использования протоколом NETBIOS.

Перед выполнением команды ее код должен быть записан в поле Cmd блока NCB. Каждая команда NETBIOS реализована в двух вариантах: с ожиданием и без ожидания окончания выполнения команды.

Все команды NETBIOS можно разделить на несколько групп.

1. Для работы с именами:

0x30, 0xB0 – добавить новое имя в таблицу имен станции (с ожиданием и без ожидания),

0x36, 0xB6 – добавить новое групповое имя в таблицу станции,

0x31, 0xB1 – удалить имя из таблицы имен станции.

2. Для приема и передачи датаграмм:

0x20, 0xA0 – передать одной или группе станций блок данных в виде датаграммы,

0x22, 0xA2 – передать всем станциям блок данных в виде датаграммы,

0x21, 0xA1 – принять блок данных, переданный с помощью команды 0x20 или 0xA0,

0x23, 0xA3 – принять блок данных, переданный с помощью команды 0x22 или 0xA2.

3. Для работы с каналами:

0x10, 0x90 – установить канал между двумя именами, заданными в блоке NCB,

0x11, 0x91 – организовать канал с вызываемой стороны (работают в паре с командами 0x10 или 0x90),

0x12, 0x92 – закрыть канал,

0x34, 0xB4 – опросить состояние канала.

4. Для приема и передачи данных через каналы:

0x14, 0x94 – передать блок данных (до 64 кб) по каналу,

0x71, 0xF1 – передать блок данных (до 64 кб) по каналу без проверки доставки блока,

0x17, 0x97 – передать два буфера (каждый по 64 кб) по каналу как один блок,

0x72, 0xF2 – передать два буфера (каждый по 64 кб) по каналу как один блок без проверки доставки блока,

0x15, 0x95 – принять блок данных, переданный по каналу,

0x16, 0x96 – принять блок данных, переданный по любому каналу, который организовала принимающая станция.

5. Другие команды:

0x32 – удалить все имеющиеся каналы и имена,

0x35 – отменить ранее запущенную команду.

Протокол NCP

В ОС NetWare протокол транспортного уровня NCP (NetWare Core Protocol) является надстройкой над протоколом сетевого уровня IPX (или IP в NetWare 5.x) и используется для организации обмена между PC и файловым сервером. Системные программы поддержки протокола NCP обеспечивают квитирование передаваемых кадров и сбор пакетов в сообщение. На рис. 8.18 представлена структура пакета NCP.

Протокол NCP реализован в NetWare 3.x на системном уровне. В NetWare 4.x предусмотрен API-интерфейс NCP Extension для обращения к протоколу NCP из прикладных программ на PC и из разрабатываемых NLM-модулей. Для обмена данными между программами по протоколу NCP используются пакеты IPX с номером гнезда 0x0451 и типом пакета 17.

Связь между PC и файловым сервером, использующих API-интерфейс к протоколу NCP, обычно организуется по следующей схеме:

NLM-модуль регистрирует какую-либо свою функцию как расширение NCP,

программа на PC или файловом сервере связывается с NetWare и получает требуемый идентификатор расширения NCP,

программа на PC или файловом сервере использует зарегистрированную функцию NLM-модуля как удаленную процедуру, передавая ей исходные данные и получая результаты обработки.

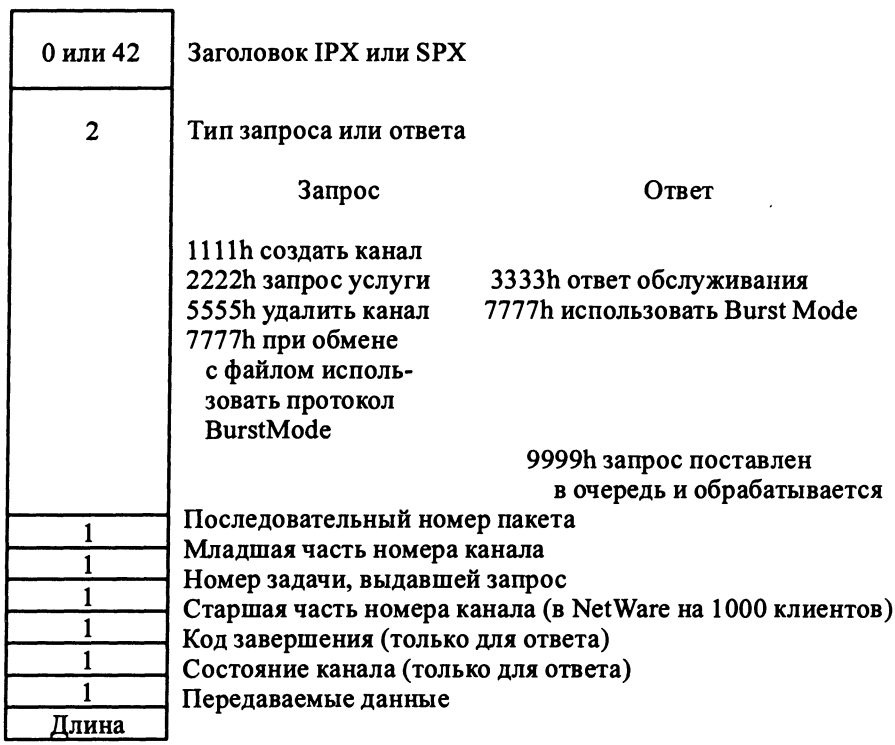


Рис. 8.18. Структура пакетов NCP

Фирма Novell предлагает API-интерфейс для работы в NetWare 4.x по протоколу NCP на PC и файловом сервере. В табл. 8.5 перечислены примитивы интерфейса для NLM-модуля (сервера).

Таблица 8.5. **Примитивы API-интерфейса для обращения к протоколу NCP из NLM-модуля**

Примитив	Описание
NWRegisterNCPExtension	Зарегистрировать функцию NLM-модуля как расширение NCP. Вход – имя регистрации, указатель на функцию (удаленную процедуру, которая выполняется как отдельная нить) и т. д. Прототип этой функции включает указатель на идентификатор соединения, указатель на буфер с исходными данными и длина этого буфера, указатель на буфер с результатами и длина этого буфера
NWDeRegisterNCPExtension	Отменить регистрацию и удалить NLM-модуль из списка расширений NCP

В табл. 8.6 перечислены основные примитивы API-интерфейса для программы (клиента), выполняемой на PC или файловом сервере.

Таблица 8.6. **Примитивы API-интерфейса для обращения к протоколу NCP из прикладной программы на PC или файловом сервере**

Примитив	Описание
NWScanNCPExtensionsInfo	Получить идентификатор расширения NCP. Вход – имя регистрации (см. табл. 8.5) и др. Выход – идентификатор расширения NCP
NWSendNCPExtensionRequest	Послать запрос на файловый сервер к удаленной процедуре. Вход – идентификатор расширения NCP, указатель на буфер с исходными данными, указатель на буфер результатов и др. Выход – результаты выполнения удаленной процедуры

Протокол TLI

Протокол TLI (Transport Layer Protocol) разработан фирмой AT&T и является надстройкой над протоколами IPX и SPX. API-интерфейс для TLI легко изучить и использовать.

Стандартный метод инициации и выполнения коммуникаций с использованием TLI состоит в следующем:

открыть конечную точку связи TLI (для IPX, SPX или TCP),
связать эту конечную точку с адресом станции,
установить связь (если используется SPX или TCP),
передать или получить данные,
отключить связь (если используется SPX или TCP).

В табл. 8.7 перечислены основные примитивы API-интерфейса для связи с протоколом TLI.

Таблица 8.7. **Основные примитивы API-интерфейса для работы с протоколом TLI**

Примитив	Описание
t_open	Открыть конечную точку TLI
t_bind	Связать конечную точку с адресом станции
t_sndudata	Послать дейтаграмму пакетного типа (как в IPX)
t_rcvudata	Получить пакет дейтаграммы
t_connect	Инициировать запрос для подключения
t_listen	Принять запрос по связи (см. t_connect)
t_accept	Послать уведомление о принятии запроса на подключение
t_rcvconnect	Получить ответ на запрос t_connect (см. t_accept)
t_snd	Послать по связи пакет информации
t_rcvdis	Отключить от связи принимающую сторону
t_snddis	Отключить от связи передающую сторону
t_look	Получить тип события, сгенерированного номером ошибки (t_error)
t_error	Отобразить сообщение об ошибке

Для поддержки протокола TLI на файловом сервере используется библиотека CLIB.NLM. На файловом сервере следует дополнительно загрузить NLM-модуль TLI.NLM для использования служб, основанных на STREAMS.

Идентификация услуг. Протокол SAP

В NetWare протокол SAP (Service Advertising Protocol) является надстройкой над протоколом IPX и используется файловым сервером для оповещения других станций о предоставляемых услугах. Для оповещения используется широковещательный пакет IPX (в поле DestNode заголовка IPX и в поле ImmAddress блока ECB указывается значение FFFFFFFFh) с номером гнезда 0x0452.

Каждый файловый сервер посылает SAP-пакет примерно через каждые 60 с (структура пакета см. рис 8.18).

С помощью этих пакетов файловый сервер информирует другие серверы о своем присутствии. Данные этого пакета помещаются в БД объектов на каждом файловом сервере сети. Если сервер не будет непрерывно объявлять свое имя, то оно теряется из БД каждого файлового сервера NetWare.

Программа PC может получить информацию о серверах сети двумя способами:

читая эти сведения в режиме Bindery из базы данных текущего файлового сервера для объектов типа 0004h (серверы NetWare),

принимая пакеты SAP, используя при этом гнездо 0x0452.

При работе Netware 5.x на «чистом» протоколе сетевого уровня IP используется протокол обнаружения служб SLP, аналогичный протоколу SAP.

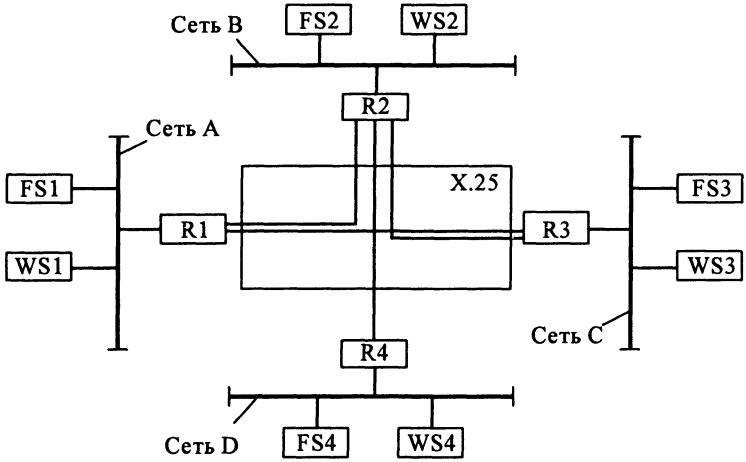


Рис. 8.19. Пример объединения сегментов ЛВС с помощью маршрутизаторов

Протоколы маршрутизации RIP и NLSP

Маршрутизатором называется специальное устройство, которое анализирует номер сети станции-получателя и направляет пакет по оптимальному маршруту. На рис. 8.19 представлен пример сети. Здесь четыре сегмента ЛВС с номерами сетей A, B, C, D связаны между собой с помощью маршрутизаторов R1, R2, R3, R4. Связь осуществляется посредством выделенных телефонных каналов, каналов связи сети X.25, Frame relay, и др.

Если PC WS1 направляет пакет на файловый сервер FS4, то маршрутизатор R1 должен выбрать оптимальный маршрут передачи: R1-R2-R4-FS4 или R1-R3-R2-R4-FS4.

В маршрутизаторах фирмы Novell используются два метода маршрутизации:

дистанционный векторный метод на базе протокола RIP (продукт NetWare MultiProtocol Router 2.1 – MPR v.2.1),

анализ состояния соединений на базе протокола NLSP (продукт NetWare MultiProtocol Router 3.x – MPR v.3.x).

Продукты MPR v.2.1 и v.3.x могут функционировать на файловом сервере или на выделенной PC.

Рассмотрим принципы работы маршрутизатора на базе MPR v.2.1 (рис. 8.20). Здесь используется протокол RIP (Routing Information Protocol) (см. § 5.5). Предположим, что на маршрутизатор R1 поступает от WS1 пакет IPX, направляемый на файловый сервер FS4 (см. рис. 8.19). R1 извлекает из заголовка пакета номер сети (D) станции-получателя и выполняет поиск строки в таблице «Сеть» (на рисунке номера сетей представлены в виде идентификаторов A, B, C, D). После этого из соответствующей строки таблицы «Канал» (в данном

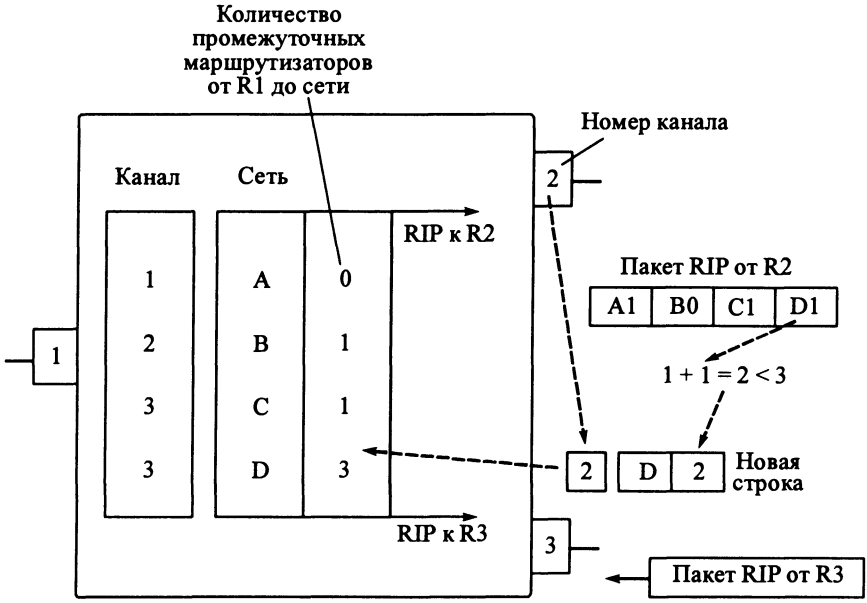


Рис. 8.20. Схема работы маршрутизатора на базе MPR v.2.1

примере из последней) читается номер канала, куда и направляется пакет. Для данного примера пакет направляется через третий канал маршрутизатору R3, где будут выполнены аналогичные действия.

В таблице «Сеть» для каждого номера сети хранится критерий передачи пакета от данного маршрутизатора до соответствующего сегмента. Значение этого критерия совпадает с числом промежуточных маршрутизаторов до сегмента. Таблица «Сеть» используется для формирования поля данных пакета RIP. Этими пакетами соседние маршрутизаторы обмениваются между собой (примерно один раз в минуту). RIP-пакет представляет собой IPX-пакет с гнездом 0x0453 и типом пакета 1.

Рассмотрим, как происходит обновление строк в таблицах «Сеть» и «Канал» маршрутизатора. Предположим, что на R1 поступил RIP-пакет от маршрутизатора R2 и значение критерия передачи данных от R2 до сети D равно 1 (это значение передается в RIP-пакете). Далее маршрутизатор R1 вычисляет новое значение критерия для передачи данных от R1 до сети D. Оно будет равно

1(промежуточный маршрутизатор R2) +
+ 1(критерий передачи данных от R2 до сети D) = 2.

Это число сравнивается со старым значением, равным 3. Так как значение нового критерия (2) меньше старого, то происходит обновление строк таблиц «Сеть» и «Канал» для сети D (рис. 8.20). Теперь пакет, передаваемый от станции WS1 на файловый сервер FS4, будет направляться маршрутизатором R1 во 2-й канал.

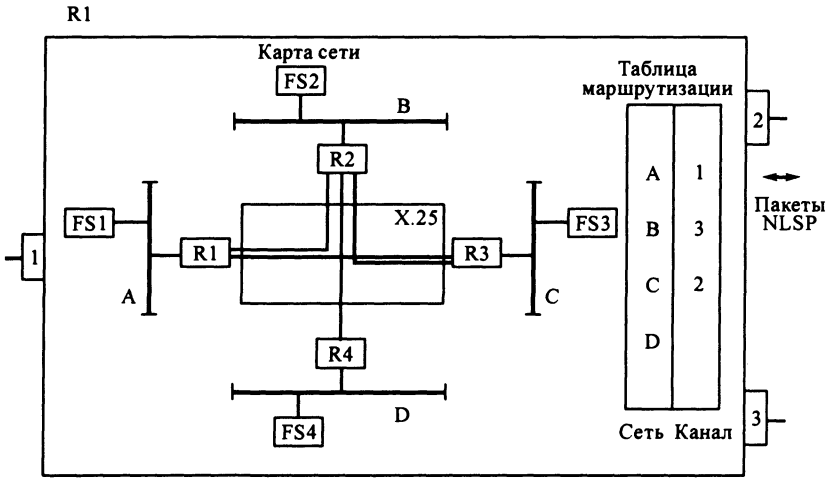


Рис. 8.21. Схема работы маршрутизатора на базе MPR v.3.x

Таким образом, RIP-пакеты обеспечивают динамическое изменение маршрутов передачи пакетов, вызванное изменением состояния сети (подключение или отключение маршрутизаторов, аварийное состояние канала передачи и т. д.). Недостатком маршрутизации на базе RIP-пакетов является широковещательный характер этих пакетов.

Рассмотрим принципы работы маршрутизатора на базе MPR v.3.x (рис. 8.21). Здесь используется протокол NLSP (Network Link Services Protocol). В начале своей работы маршрутизатор выполняет опрос узлов сети и автоматически строит карту сети на уровне маршрутизаторов и файловых серверов сети. Можно вручную задать цену каждой связи. На основе этой карты маршрутизатор формирует оптимальную таблицу маршрутизации.

Предположим, что от WS1 на R1 поступает пакет IPX, направляемый на файловый сервер FS4. Маршрутизатор извлекает из заголовка пакета номер сети станции-получателя, выполняет поиск строки в таблице маршрутизации (для нашего примера D, 2) и направляет пакет в соответствующий канал (в данном случае 2).

После поступления NLSP-пакета маршрутизатор выполняет обновление карты сети и таблицы маршрутизации. Карта сети строится и хранится каждым маршрутизатором, что делает ненужным постоянный широковещательный обмен маршрутной информацией до тех пор, пока не будет изменена конфигурация сети. По утверждению специалистов Novell этот более эффективный метод информационного обмена между маршрутизаторами может обеспечить снижение сетевого трафика до 40 %.

8.3. Расширяемость и открытость

Системные сетевые NLM-приложения

Выше отмечалось, что в качестве NLM-модулей выступают драйверы жестких дисков (*.DSK, для версии 5.x – *.HAM и *.CDM), драйверы сетевых адаптеров (*.LAN), модули поддержки пространства имен (*.NAM), программы с расширением *.NLM. Все модули с расширением NLM можно условно разделить на несколько групп:

системные библиотеки NLM (STREAMS.NLM, CLIB.NLM, TLI.NLM, SPXS.NLM, IPXS.NLM и т. д.),

системные утилиты файлового сервера (табл. 8.8),

модули, расширяющие возможности NetWare: модули шлюза электронной почты MHS, модули маршрутизатора MPR, модули коммуникационных серверов NACS и NetWare Connect, модули серверов БД, модули объединения серверов в кластеры, модули для связи с другими ОС и т. д.

В табл. 8.8 перечислены основные системные утилиты файлового сервера, поставляемые с дистрибуцией NetWare, хранящиеся в каталоге SYS:SYSTEM и загружаемые по команде LOAD.

Некоторые NLM-модули, обеспечивающие работу шлюза MHS, маршрутизатора MPR для файлового сервера, кластера серверов, входят в стандартную поставку NetWare 4.x. Существует большое число других NLM-модулей, расширяющих возможности NetWare. Их можно приобрести за дополнительную плату. Количество таких программ постоянно растет.

Поддержка рабочих станций разных платформ

NetWare поддерживает связь с PC, на которых установлены ОС MS DOS, Windows 95/98/NT, Macintosh, OS/2, UNIX (рис. 8.22). На каждой PC должно быть установлено свое ПО клиента. Структура этого ПО рассмотрена ниже.

NetWare поддерживает форматы, отличные от DOS. Файлы ОС Windows 95/98/NT (OS/2), Macintosh, UNIX, которые загружаются на PC, имеют другие наборы атрибутов, длины имен файлов и т. д. Чтобы поддержать работу таких

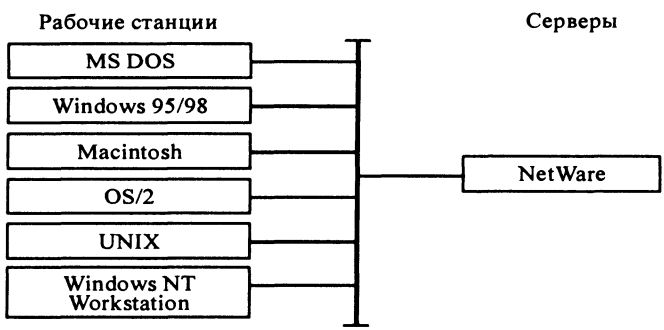


Рис. 8.22. Рабочие станции, поддерживаемые NetWare

Таблица 8.8. **Утилиты файлового сервера NetWare**

Имя NLM-модуля	Описание
NetWare 3.x , 4.x, 5.x	
EDIT.NLM	Отредактировать с консоли текстовый файл, хранящийся в каталоге DOS или NetWare файлового сервера
REMMOTE.NLM и RSPX.NLM	Организовать удаленную консоль. На PC необходимо запустить утилиту RCONSOLE.EXE
INSTALL.NLM	Установить или модифицировать характеристики ОС NetWare
PSERVER.NLM	Запустить сервер печати
VREPAIR.NLM	Исправить логическую структуру данных сетевого тома после сбоя (т. е. «отремонтировать» таблицы DET и FAT)
MONITOR.NLM	Посмотреть параметры текущего состояния сети
NetWare 4.1	
SERVMAN.NLM	В диалоге посмотреть и установить SET-параметры
DOMAIN.NLM	Создать ОС-защищенный домен (OS_PROTECTED). Загрузку этого модуля можно кодировать только в файле STARTUP.NCF
NetWare 4.x, 5.x	
KEYB.NLM	Изменить кодовую таблицу клавиатуры. При запуске этого модуля с параметром RUSSIA устанавливается переключатель для ввода с консоли латинских или русских букв
DSREPAIR.NLM	Отремонтировать дерево NDS и его реплики
CDROM.NLM	Используя затем команды CD, можно смонтировать CD-ROM как том NetWare. Предварительно должен быть загружен драйвер для работы с CD-ROM, который должен быть подключен по SCSI-интерфейсу. На PC для планирования драйва на том CD-ROM следует использовать утилиту MAP
RPL.NLM	Поддержать загрузку ОС рабочей станции в оперативную память бездисковой станции
TIMESYNC.NLM	Организовать мониторинг времени на сервере NetWare 4.x (загружается автоматически)

станций, на файловом сервере должны быть загружены различные пространства имен. *Пространство имен* представляет собой дополнительную запись таблицы DET (рис. 8.23). Таким образом, на томе с активными пространствами имен Macintosh, UNIX, Windows 95/98/NT (OS/2) будут храниться четыре

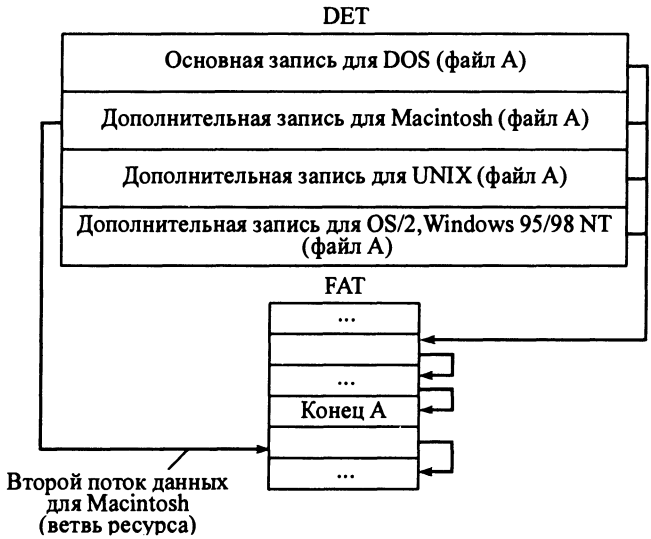


Рис. 8.23. Организация пространств имен файлового сервера

записи для каждого файла: основная запись каталога и записи каталогов для *Ma*cintosh, UNIX, Windows 95/98/NT (OS/2). Все записи ссылаются на одну и ту же цепочку элементов FAT (поток данных), т. е. физически файл записывается на диск один раз. На Macintosh файлы хранятся с использованием двух потоков данных (ветвей). Одна ветвь содержит информацию о ресурсе Macintosh для этого файла (ветвь ресурсов), а другая – фактические данные.

Каждое пространство имен поддерживается своим NLM-модулем с расширением NAM: MAC..NAM – для Macintosh, LONG.NAM – для Windows 95/98/NT и OS/2 и NFS.NAM – для UNIX. Чтобы добавить необходимые записи в таблицы DET и FAT тома, с консоли файлового сервера необходимо для каждого пространства имен выполнить один раз команду

ADD NAME SPACE имя TO том

Здесь «имя» – это или MAC, или LONG, или NFS. Для дальнейшей работы достаточно загружать только соответствующие NLM-модули поддержки пространства имен.

Многопротокольный интерфейс ODI рабочей станции

На рис. 8.24 представлена структура программного обеспечения клиента NetWare, устанавливаемого на PC. Запросчик (Requester) перехватывает прерывания на ввод/вывод в файл. Если файл располагается на локальном диске, то запрос обрабатывается ОС рабочей станции. Иначе, используя примитивы API-интерфейса, запросчик организует ввод/вывод в файл, хранящийся на диске файлового сервера. В OS/2 прерывания перехватывает сама ОС, а затем передает управление запросчику.

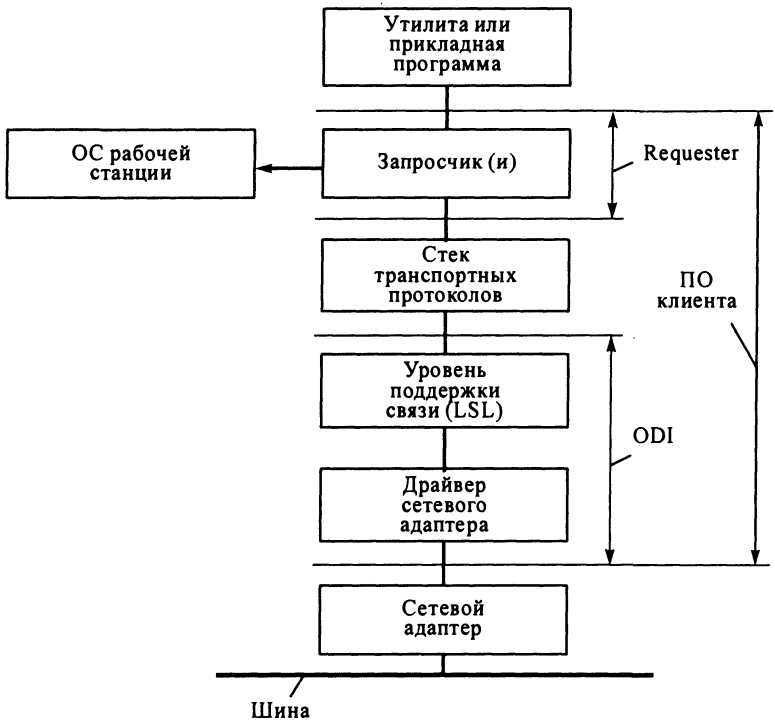


Рис. 8.24. Структура ПО клиента NetWare

Стек транспортных протоколов поддерживает API-интерфейс доступа запросчика (или прикладной программы) к службам NetWare. На PC могут выполняться несколько транспортных протоколов одновременно: SPX/IPX, TCP/IP, AppleTalk. Как правило, для каждого протокола требуется свой запросчик, т. е. свой API-интерфейс. Обычно протокол SPX/IPX используют клиенты MS DOS, OS/2, Windows 95/98/NT, протокол AppleTalk – клиенты Macintosh. Протокол TCP/IP устанавливается в стек клиента NetWare для обеспечения работы клиента по этому протоколу и для связи с ОС UNIX.

Программа уровня поддержки связи (LSL – Link Support Layer) принимает пакет от драйвера сетевого адаптера, распознает протокол, который был использован при формировании пакета, выбирает соответствующий протокол из стека и передает ему управление. Модули драйвера и уровня поддержки связи со стеком протоколов образуют так называемый интерфейс ODI (Open Data-Link Interface). Важно отметить, что стек транспортных протоколов PC является открытым: если новый протокол разрабатывался с использованием спецификаций ODI-интерфейса, то он может быть включен в стек.

В табл. 8.9 перечислены компоненты ПО 16-разрядного клиента MS DOS.

Таблица 8.9. **Компоненты программного обеспечения клиента MS DOS**

Запросчик	Транспортный протокол (SPX/IPX)	Уровень поддержки связи	Драйвер сетевого адаптера
VLM.EXE и модули *.VLM	IPXODI.COM	LSL.COM	NE2000.COM, NE1000.COM и др.

Многопротокольный интерфейс STREAMS файлового сервера

На рис. 8.25 представлена структура многопротокольного интерфейса STREAMS, используемого на файловом сервере. Его применяют для обеспечения взаимодействия с сервером по протоколам, отличным от NCP на базе IPX. Поддержка многопротокольного интерфейса STREAMS – это множество программных средств, содержащих ресурсы ядра и NLM-библиотеки STREAMS.NLM, CLIB.NLM, TLI.NLM, SPXS.NLM, IPXS.NLM. Это средство используют, чтобы создать дуплексный процесс пересылки данных между драйверами и программами сервера.

Драйверы сетевых адаптеров – это NLM-модули, обеспечивающие прием кадров и выделение из них пакетов. Файловый сервер может поддерживать одновременную работу 16 драйверов.

Уровень поддержки связи (LSL – Link Support Layer) принимает пакет от драйвера, распознает тип пакета и выбирает соответствующий протокол из стека

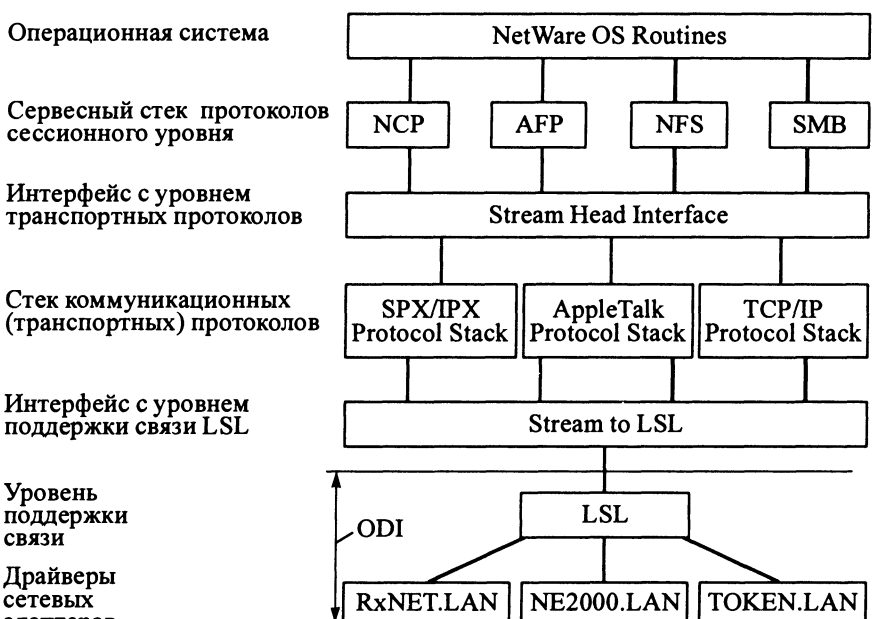


Рис. 8.25. Структура многопротокольного интерфейса STREAMS на файловом сервере

коммуникационных протоколов. NetWare поддерживает одновременную работу до 32 протоколов. Модули драйверов и уровня поддержки связи LSL образуют ODI-интерфейс, позволяющий включать в стек новые транспортные протоколы. При необходимости пакеты обрабатываются протоколами сессионного уровня NCP (пакеты IPX), AFP (пакеты AppleTalk от клиентов Macintosh), NFS (пакеты TCP/IP от клиентов UNIX), SMB (пакеты IPX от клиентов OS/2).

Взаимодействие NetWare с другими сетевыми ОС

Рассмотрим принципы организации взаимодействия NetWare с ОС Windows NT и UNIX, которые являются ее основными конкурентами. Еще в 1994 г. фирма Microsoft объявила о выпуске ряда продуктов, которые облегчают взаимодействие серверов Windows NT и NetWare в сети или обеспечивают переход с NetWare на серверы Windows NT. Эти продукты перечислены в табл. 8.10.

Таблица 8.10. **Продукты Microsoft, обеспечивающие взаимодействие серверов Windows NT и NetWare**

Продукт или утилита	Описание
Набор программ, реализующих протокол IPX, IPX-Compatible Transport Stack	Позволяет устанавливать и использовать приложения, ориентированные на Windows NT Server, не изменяя ПО клиента NetWare
Шлюзовая служба Gateway Service for NetWare	Дает возможность использовать Windows NT Server в качестве коммуникационного средства (шлюза) и предоставляет Windows NT-станциям доступ к серверам NetWare. Шлюз транслирует SMB-пакеты, посланные с Windows NT-станции (Windows for Workgroups, Windows NT Workstation), в NCP-пакеты и пересылает их серверу NetWare
Средство переноса Migration Tool for NetWare	Автоматически переносит с сервера NetWare на сервер Windows NT Server информацию о бюджетах пользователей, сценарии регистрации при входе в систему, файлы, каталоги, средства защиты
Службы файлов и печати File and Print Services for NetWare	Размещает службы каталогов и печати системы NetWare и деловые приложения на одном компьютере под управлением Windows NT Server, не требуя при этом изменения ПО клиента NetWare и конфигурации сети. Для клиента NetWare сервер Windows NT Server превращается в NCP-сервер
Менеджер службы каталогов Directory Service Manager (DSM) for NetWare	Использует сервер Windows NT Server для централизованного управления бюджетами пользователей в смешанной сети, работающей и с NT, и с NetWare (2.x и 3.x), что позволяет конечным пользователям однократно регистрироваться при входе в систему и иметь единый бюджет и пароль

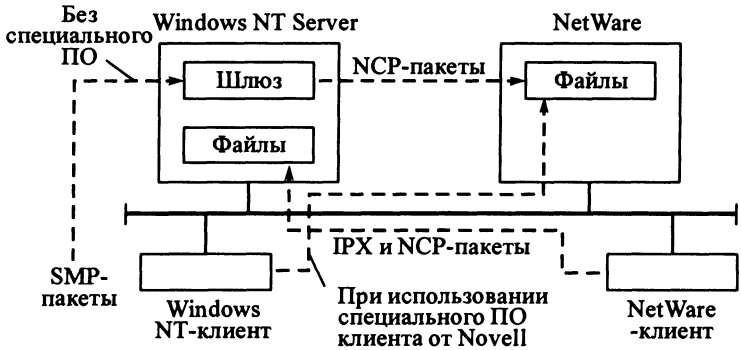


Рис. 8.26. Схема взаимодействия клиентов с ОС NetWare и Windows NT Server

На рис. 8.26 представлена схема взаимодействия Windows NT-клиентов и NetWare-клиентов (работающих по протоколу IPX) соответственно с ОС NetWare и Windows NT Server, функционирующих в одной сети.

Рассмотрим, как обеспечивается взаимодействие NetWare с ОС UNIX. Для этого можно использовать продукт NetWare NFS 1.2c. Он функционирует как совокупность NLM-модулей, устанавливаемых на файловом сервере NetWare (табл. 8.11).

Таблица 8.11. **NLM-модули NetWare NFS 1.2c, обеспечивающие взаимодействие NetWare с UNIX**

NLM-модуль	Описание
NLM NFS Server	Позволяет UNIX-клиентам работать с файловой системой NetWare как с расширением локальной файловой системы UNIX (поддержка распределенной файловой системы)
NLM Lock Manager	Выполняет блокировку записей и файлов NetWare (в среде NFS)
NLM File Transfer Protocol (FTP)	Позволяет FTP-клиентам инициировать вывод файлов на сервер NetWare и читать их с сервера
NLM NFS NameSpace	Позволяет поддерживать на сервере NetWare пространство имен для UNIX
NLM Line Printer Daemon	Реализует для UNIX-клиентов механизм передачи заданий в очереди печати NetWare. Пользователи UNIX для доступа к подключенным к NetWare принтерам могут использовать команды UNIX (lp, lpq, lprm)
NLM Print Gateway	Позволяет NetWare-клиентам для буферизации заданий печати на принтерах, подключенных к UNIX, применять команды NetWare (nprint, capture и т. д.)

Окончание табл. 8.11

NLM-модуль	Описание
NLM XCONSOLE	Реализует удаленное администрирование сервера NetWare с любой станции UNIX-клиента, поддерживающей VT100, VT220 или X Windows System

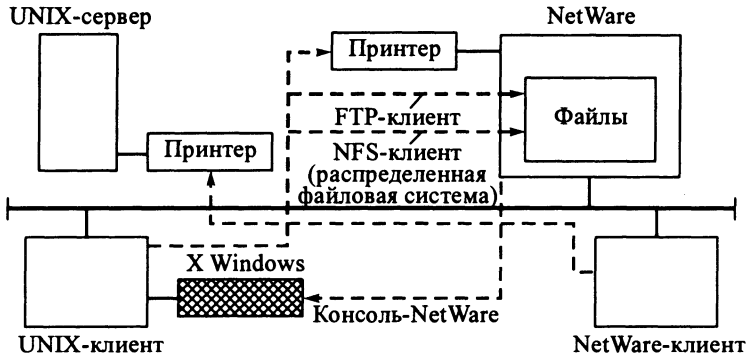


Рис. 8.27. Схема взаимодействия клиентов с ОС NetWare и UNIX

На рис. 8.27 представлена схема взаимодействия (на базе NetWare NFS 1.2c) UNIX-клиентов и NetWare-клиентов соответственно с ОС NetWare и UNIX-сервером, функционирующих в одной сети. Так как средство NetWare NFS 1.2c не обеспечивает доступ клиентов NetWare к файловой системе UNIX, то для этой цели можно использовать следующие продукты:

- LAN WorkPlace или LAN WorkGroup (Novell), обеспечивающие включение протокола TCP/IP в стек протоколов PC;
- PC Protocol Services for Solaris (SunSoft), обеспечивающий включение IPX в стек протоколов на UNIX-сервере с ОС Solaris.

8.4. Обеспечение высокой производительности

«Плоская» модель основной памяти

Как уже отмечалось, компилятор Watcom C генерирует код, использующий преимущества архитектуры процессоров. Он использует 32-битовые ближайшие указатели (near pointers) и 4-байтовые целые числа. Использование 32-битовых указателей делает концепцию моделей памяти (Tiny, Small, Compact, Medium, Large, Huge и т. д.) во многом ненужной. NLM-модули компилируются для непрерывной модели памяти с абсолютной адресацией («плоской» модели). 32-битового указателя достаточно для адресации всей доступной памяти. Кроме того, компилятор генерирует код, выполняемый в виртуальном режиме (для NetWare 4.x/5.x). В NetWare 4.x/5.x используется страничная организация памяти.

Таким образом, основная память файлового сервера NetWare интерпретируется как один большой сегмент, но идентифицируется для использования NLM с помощью функций распределения. Когда процесс (нить) запрашивает память, ему выделяется пул памяти. Этот пул может быть освобожден, но оставлен в пуле процесса, либо освобожден и возвращен в системный пул.

Когда запрашивается память, NetWare использует три массива указателей для определения того, где находится доступная память. Это массивы отслеживают блоки доступной памяти. Первый массив отслеживает блоки с 16-байтовыми приращениями от 16 б до 1024 б каждый. Второй массив отслеживает блоки с 256-байтовыми приращениями. Третий массив указателей отслеживает блоки, превышающие 4 кб. Основываясь на размере запрошенной NLM-модулем памяти, NetWare выполняет поиск в соответствующем массиве, пока не находит затребованный объем памяти.

В NetWare 4.x/5.x внесены улучшения:

• Когда память выделяется и освобождается снова и снова, это может привести к ее фрагментации, и некоторые блоки будут оставаться неиспользуемыми. В таких случаях выполняется некоторая работа по очистке. Такая «сборка мусора» обеспечивает использование ранее недоступных блоков памяти. В табл. 8.12 перечислены SET-параметры NetWare 4.x/5.x, регулирующие «сбор мусора».

Таблица 8.12. **SET-параметры, регулирующие «сбор мусора» в NetWare 4.x/5.x**

SET-параметр	Значение по умолчанию	Границы изменения	Примечания
Garbare Collection Interval	15 мин	1 мин....1 ч	Определяет максимальный интервал времени между «чистками» памяти
Number of Frees for Garbare	1000	100...10000	Определяет минимальное число освобождений памяти для запуска системной программы чистки памяти
Minimum Free Memory for Garbare Collection	8000	1000...1000000	Определяет минимальный размер освобождаемой памяти для запуска программы чистки памяти

• Предыдущие версии NetWare имели несколько пулов памяти, с которыми должен был работать программист, создающий NLM-модуль. В NetWare 4.x/5.x он может работать с одним системным пулом. После того, как память будет получена из системного пула, каждый NLM-модуль работает со своим собственным пулом памяти, пока память не возвращается в системный пул.

Невытесняющая многозадачность

Обработка, которая встречается в многозадачных сетевых ОС, основывается на одной из двух моделей использования центрального процессора (ЦП) –

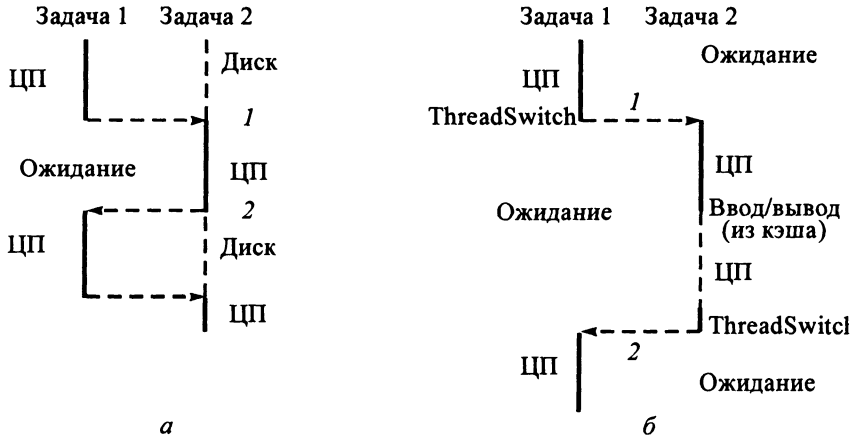


Рис. 8.28. Схемы переключения процессов:
а – для модели с приоритетами; *б* – для модели без приоритетов

модель с приоритетами и модель без приоритетов (невытесняющая многозадачность). Большинство многозадачных сетевых ОС, таких, как OS/2 и UNIX, являются системами, где поддерживается модель с приоритетами. Предположим, что в одной из этих ОС выполняются две задачи (рис. 8.28, *а*), причем задача 2 имеет приоритет выше, чем задача 1.

После операции с диском (чтение или запись) ОС активизирует задачу 2, имеющую более высокий приоритет (рис. 8.28, *а*, точка *1*). При этом задача 1 прерывается (вытесняется) и переходит в состояние ожидания. Если для задачи 2 требуется выполнить операцию с диском, то ОС приостановит ее и передаст управление задаче 1 (рис. 8.28, *а* точка *2*) и т. д.

Многозадачная сетевая ОС NetWare является системой, где поддерживается модель без приоритетов (невытесняющая многозадачность). Если какая-либо задача (нить) выполняет функцию ThreadSwitch, то ОС помещает ее в конец очереди RunList и передает управление другой задаче (рис. 8.28, *б* точки *1*, *2*).

В системах с приоритетами необходимо перед обновлением ресурсов (областей основной памяти, записей файлов и т. д.) выполнить их блокировку (рис. 8.29, *а* точки *1*, *2*, *3*, *4*).

Это связано с тем, что задача может быть прервана в любой момент времени. Но, во-первых, на блокировку и разблокировку разделяемых ресурсов тратится процессорное время, что снижает производительность системы. Во-вторых, при использовании блокировок часто возникают тупиковые ситуации.

Предположим, что задача 1 заблокировала запись 1 какого-либо файла (рис. 8.29, *а* точка *1*) и была прервана задачей 2. Пусть задача 2, в свою очередь, блокирует запись 2 (точка *2*). При попытке заблокировать запись 1 задачей 2 она переходит в состояние ожидания, так как эта запись уже заблокирована задачей 1. Управление передается задаче 1, которая пытается блокировать запись 2 и переходит в состояние ожидания, так как эта запись уже была заб-

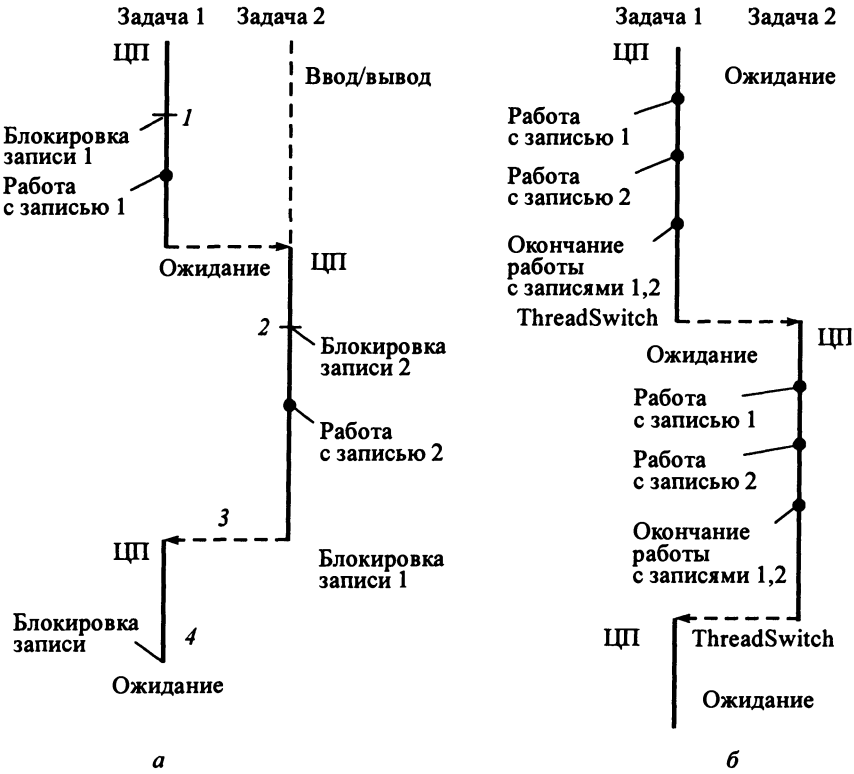


Рис. 8.29. Схемы использования разделяемых ресурсов:
а – для модели с приоритетами; *б* – для модели без приоритетов

локирована задачей 2. Таким образом, ни одна из задач (1 и 2) не может продолжить выполнение из-за возникшей тупиковой ситуации.

Преимуществом системы с приоритетами является то, что при выполнении операции с диском процессор переключается на выполнение другой задачи.

В NetWare, где поддерживается модель использования процессора без приоритетов, переключение на другую задачу планирует сама программа (NLM-модуль), используя команду ThreadSwitch в подходящий момент, т. е. задача 1 может выполнить все требуемые обновления в разделяемых записях 1 и 2, а затем передать управление другой нити с помощью команды ThreadSwitch (рис. 8.29, *б*). В этом случае нет необходимости использовать блокировки ресурсов. Это, во-первых, повышает быстродействие системы, а, во-вторых, устраняет возможность возникновения тупиковых ситуаций.

Недостатком системы без приоритетов является то, что при выполнении операции с диском (чтение в кэш с диска) ОС NetWare не переключает процессор на выполнение другой прикладной задачи. Но этот недостаток компенсируется тем, что в NetWare, как правило, используется кэш-память большого размера и 90 % запросов на ввод данных удовлетворяется из этого кэша.

Описанный выше механизм невытесняющей многозадачности использовался в NetWare 3.x и 4.x. В NetWare 5.x реализована смешанная стратегия. По умолчанию задача может асинхронно вытесняться из процессора и оперативной памяти задачей, которая имеет более высокий приоритет. Но программист с помощью специальных API-функций может отметить участок программы, где будет действовать режим невытесняющей многозадачности.

Тотальная буферизация файлового ввода/вывода

Выше была рассмотрена структура ОП файлового сервера NetWare и схема формирования кэш-памяти (кэш-буфера). NetWare кеширует данные файла поблочно. Это позволяет файловой системе NetWare поддерживать тесную синхронизацию между кэш-буфером и физической дисковой памятью, что помогает обеспечить целостность данных файла и дает большой выигрыш в производительности.

Рассмотрим алгоритм работы NetWare с кэш-памятью при чтении и обновлении блоков данных диска (рис. 8.30). При выполнении функции чтения данных из файла сервера ОС NetWare рассчитывает адрес требуемого блока на диске и проверяет, находится ли он в кэше. Если да, то данные пересылаются из буфера кэша в пул NLM-модуля, выдавшего запрос на чтение. Если требуемого блока нет в кэше и здесь имеется свободный буфер, то блок читается в этот буфер. Если свободных буферов нет, то ОС выполняет поиск буфера, который наиболее длительное время не использовался (алгоритм LRU) и перезаписывает его на диск, если он был отмечен как «грязный» (dirty). На место перезаписанного буфера читается требуемый блок.

При выполнении функции обновления данных какого-либо файла сервера ОС читает при необходимости требуемый блок в кэш-память (см. выше), выполняет операцию обновления и отмечает этот буфер как «грязный». Обновленный буфер попадает на диск не сразу. ОС через определенный интервал времени запускает системный процесс, который анализирует кэш-память и перезаписывает «грязные» буфера на диск. Интервал времени, через который запускается системный процесс, регулируется с помощью двух SET-параметров (табл. 8.13).

Рис. 8.30. Организация работы с кэш-памятью

Таблица 8.13. **SET-параметры, регулирующие интервал перезаписи «грязных» буферов на диск**

SET-параметр	Значение по умолчанию, с	Границы изменения, с	Примечания
Dirty Directory Cache Delay Time	0,5	0 ...10	Определяет, через какой интервал времени перезаписываются на диск «грязные» буфера директорий, где хранятся записи таблиц DET
Dirty Disk Cache Delay Time	3,3	0,1 ... 10	Определяет, через какой интервал времени перезаписываются на диск «грязные» буфера файлов

Упорядочивание и распараллеливание запросов поиска на дисках

В настоящее время наибольшее распространение получили два интерфейса связи контроллера с жестким диском:

IDE/ATA и все производные от него интерфейсы Fast ATA-2, Ultra DMA/33 (скорость 16.6 Мб/с),

Fast SCSI-2 (10 Мб/с), Fast/Wide SCSI-2 (20 Мб/с), Ultra SCSI (20 Мб/с), Ultra/Wide SCSI (40 Мб/с), Ultra2 SCSI (80 Мб/с).

Интерфейс IDE поддерживает два канала, к каждому из которых могут подключаться по два устройства (HARD-диски, CD-ROM и т. д.). Контроллеры IDE/ATA перекладывают выполнение значительной части низкоуровневых задач на центральный процессор, тем самым снижая общую производительность сервера. Практически все диски с интерфейсом IDE имеют частоту вращения шпинделя 5400 об/мин, что обеспечивает скорость дисковых операций 3...5 Мб/с.

SCSI-контроллеры (рис. 8.31) сами выполняют большинство низкоуровневых операций ввода/вывода, при этом они оптимизируют запросы к периферийному оборудованию (лифтовый поиск и др.). Частота вращения шпинделя дисков SCSI составляет 7200 об/мин, что соответствует скорости обмена 4...7 Мб/с. Компании Seagate и IBM выпустили дисководы с частотой вращения 10000 об/мин, установив новую планку производительности (6...9 Мб/с).

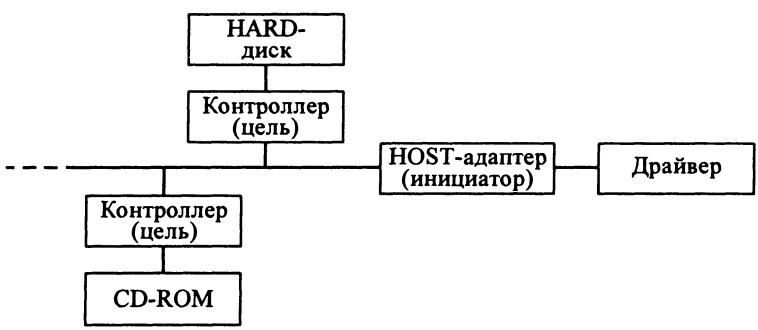


Рис. 8.31. Схема подключения устройств по интерфейсу SCSI

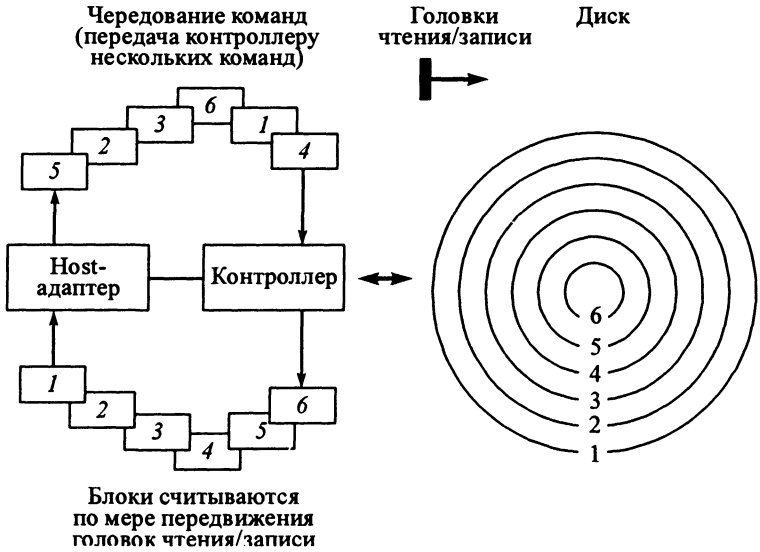


Рис. 8.32. Восходящий (лифтовый) поиск на диске

Для подключения какого-либо устройства по интерфейсу SCSI в расширительный слот компьютера (в частности файлового сервера) устанавливается HOST-адаптер (HBA, DCB и т. д.). К HOST-адаптеру можно подключить до 7 контроллеров (2 внутренних и 5 внешних): HARD-диски, CD-ROM, CD-R, принтеры, сетевые адаптеры, сканеры текстов, стриммеры. Обмен данными с подключенными устройствами выполняется в режиме мультиплексирования (т. е. параллельного доступа).

Для HARD-дисков, подключенных по SCSI-интерфейсу, поддерживается восходящий поиск (интеллект шины) на аппаратном уровне (рис. 8.32). По запросу прикладной программы драйвер жесткого диска рассчитывает номер цилиндра, номер поверхности и номер блока, где располагаются требуемые данные, и формирует команду, направляемую HOST-адаптеру.

Рабочие станции взаимодействуют с драйвером жесткого диска файлового сервера по NCP-протоколу. Поэтому на HOST-адаптер может поступить несколько команд одновременно. В этом случае HOST-адаптер направляет эти команды контроллеру диска (см. рис. 8.32). На рисунке цифрами обозначены номера цилиндров, указанных в командах поиска. Гребенка головок чтения/записи начинает движение с нулевого цилиндра, и требуемые блоки читаются не в последовательности их указания в командах, а в последовательности их размещения на цилиндрах, т. е. сначала будут считаны блоки, расположенные на 1-м цилиндре, затем на 2-м цилиндре и т. д. В этом случае гребенка головок выступает в роли лифта, цилиндры – в роли этажей, блоки – в роли пассажиров.

Использование этого метода чтения данных существенно повышает производительность дисковой системы. Действительно, если бы блоки читались в

той последовательности, в которой они были указаны в командах, то для этого потребовалось бы 19 перемещений головок (для примера на рис. 8.32). При использовании лифтового поиска для этого потребуется 6 перемещений.

При использовании интерфейса IDE ОС NetWare выполняет программное моделирование восходящего поиска, который был рассмотрен выше.

Протоколы передачи Packet Burst Protocol и LIP

В штатной конфигурации NetWare 3.11 каждый переданный кадр квитируется принимающей стороной (рис. 8.33). В NetWare 3.12/4.x/5.x используется протокол Packet Burst Protocol, позволяющий без квитирования читать/писать данные объемом до 64 кб. Размер окна (количество кадров, передаваемых без квитирования) является переменным и зависит от объема передаваемых данных и максимального размера пакета. Так, при передаче данных объемом 64 кб при размере пакета 512 б потребуется 128 кадров, а при размере пакета 1500 б – 44 кадра. В квитанции на окно указываются номера кадров, которые были приняты с ошибкой. Передающая станция должна повторить передачу этих ошибочных кадров. При увеличении числа кадров, принятых с ошибкой, NetWare автоматически уменьшает размер окна. Использование протокола Packet Burst Protocol позволяет существенно уменьшить трафик сети и тем самым повысить производительность системы.

В NetWare 3.11 протокол Packet Burst Protocol является нештатным средством и его следует инициировать на PC (BNETx.COM) и файловом сервере (PBURST.NLM). В NetWare 3.12/4.x/5.x этот протокол включается автоматически. Его нельзя выключить на сервере (для NetWare 3.12/4.x/5.x), но можно отключить для отдельной PC. Чтобы включить или отключить этот протокол на PC, в конфигурационный файл (например NET.CFG) необходимо добавить строку

PB BUFFERS = n,

где n = 0 – 10 – число буферов (а не размер окна), выделенных на PC. Если указано n = 0, то протокол Packet Burst Protocol отключается для этой PC.

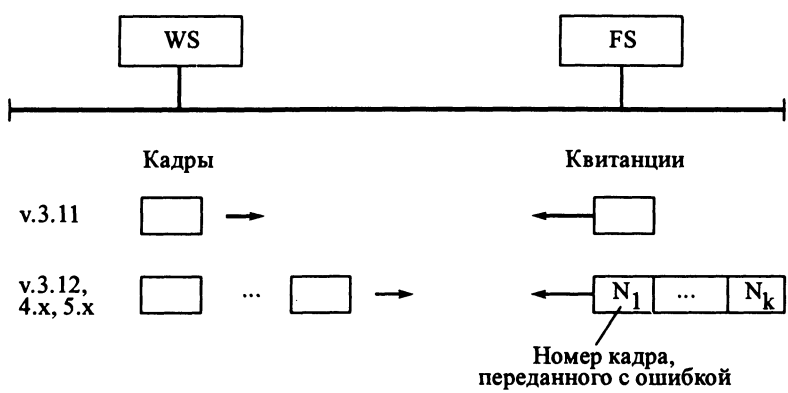


Рис. 8.33. Квитирование кадров

Если при согласовании максимальной длины пакетов, передаваемых между PC и файловым сервером, сервер обнаруживает на своем пути маршрутизатор и он не поддерживает протокол LIP (Large Internet Packet), то максимальная длина этих пакетов будет равна 576 б. Использование протокола LIP позволяет устранить этот недостаток и увеличить максимальный размер передаваемого пакета.

Для использования протокола LIP в NetWare 3.11 необходимо на PC загрузить модуль BNETx.COM, на файловом сервере – LIPX.NLM. В NetWare 3.12/4.x/5.x можно сбросить признак применения LIP-протокола с помощью SET-параметра

ALLOW LIP = OFF (для NetWare 4.x/5.x)

или

0FF5 (для NetWare 3.12)

файлового сервера или параметра

LARGE INTERNET PACKETS = OFF

в разделе NetWare DOS Requester файла NET.CFG на PC.

Максимальная длина пакета обмена с маршрутизатором устанавливается SET-параметром

MAXIMUM PHYSICAL RECEIVE PACKET SIZE = размер

файлового сервера и параметром

LIP START SIZE = размер

в разделе NetWare DOS Requester файла NET.CFG на рабочей станции.

Для сетей Token Ring и ARCnet этот максимальный размер пакета равен 4202 байтов, для сетей Ethernet – 1500 б. Эти размеры должны быть также установлены на всех маршрутизаторах сети.

8.5. Обеспечение надежности

NetWare обеспечивает надежное хранение данных на файловом сервере. Фирма Novell условно разделила соответствующие средства поддержки надежности на три уровня защиты SFT (System Fault Tolerance): SFT-I, SFT-II, SFT-III.

Первый уровень защиты SFT-I

Первый уровень защиты SFT-I включает следующие решения:

- дублирование таблиц DET- и FAT-тома,
- проверку записи на диск последующим чтением,
- динамическую переадресацию блоков (Hot Fix).

Рассмотрим эти решения подробнее.

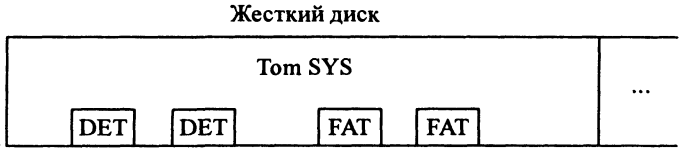


Рис. 8.34. Дублирование таблиц DET- и FAT-тома

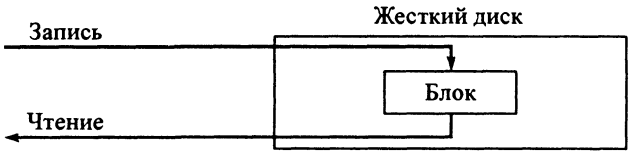


Рис 8.35. Проверка записи последующим чтением

Дублирование таблиц DET и FAT тома. Таблицы DET- и FAT-тома дублируются для каждого тома файлового сервера (рис. 8.34). Дубли используются NLM-модулем VREPAIR.NLM в процессе «ремонта» тома. Программа этого модуля последовательно читает записи DET и FAT, сопоставляет их, при необходимости корректирует эти записи и затем записывает изменения в обе копии таблиц.

Проверка записи на диск последующим чтением. Если этот режим включен, то после записи на диск блок данных читается и данные сравниваются с тем, что было записано (рис. 8.35).

Считается, что запись на диск выполнена успешно, если записанные и прочитанные данные совпадают. Режим проверки записи последующим чтением можно включить для всех дисков файлового сервера с помощью SET-параметра ENABLE DISK READ AFTER WRITE. Для конкретного диска этот режим можно включить/выключить с помощью утилиты MONITOR.NLM (пункт меню Disk Information/Read After Write Verify).

Динамическая переадресация дефектных блоков (Hot Fix). Если записанные и затем прочитанные данные не совпадают, то считается, что соответствующая область на диске является дефектной и блок записывается в область переназначения Hot Fix (рис. 8.36).

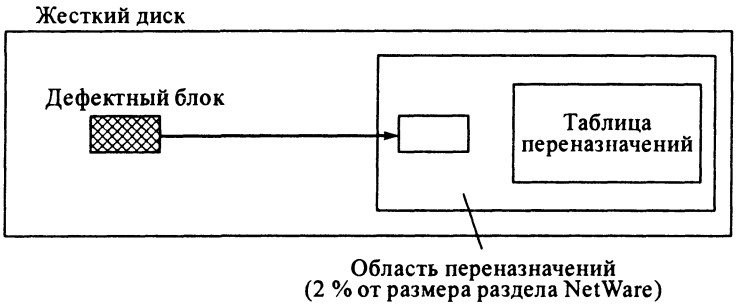


Рис. 8.36. Динамическая переадресация дефектных блоков

При чтении блока с файлового сервера сначала ОС NetWare просматривает таблицу переназначения, и если блок описан в этой таблице, то он читается из области переназначения Hot Fix. В противном случае блок читается из основной области раздела NetWare. Размер области Hot Fix устанавливается при инсталляции файлового сервера.

Второй уровень надежности SFT-II

Второй уровень надежности SFT-II включает следующие решения:

- зеркальное отображение дисков (Disk Mirroring);
- дублирование дисков (Disk Duplexing);
- систему отслеживания транзакций TTS (Transaction Tracking System);
- использование устройств бесперебойного питания (UPS).

Рассмотрим эти решения подробнее.

Зеркальное отображение дисков. Ранние модели интерфейса IDE с жестким диском включают один канал, к которому могут быть подключены два диска. С помощью программы INSTALL.NLM можно выполнить зеркализацию диска 1 на диск 2 (рис. 8.37). Диск 1 является основным, а диск 2 – вспомогательным. После записи блока на диск 1 ОС NetWare автоматически записывает тот же блок на диск 2 (выполняет зеркальное отображение).

При вводе блок читается с диска 1. Если при чтении произошла неустранимая ошибка ввода/вывода, то NetWare помещает сообщение о неустранимой ошибке ввода/вывода в журнал администратора и отображает это сообщение на экране файлового сервера. При этом NetWare автоматически переключается на работу с диском 2. Для замены диска 1 администратор должен выполнить следующие действия:

размонтировать тома диска 1,

отменить зеркальное отображение с помощью утилиты INSTALL.NLM (в NetWare 5.x – NWCONFIG.NLM),

выйти из системы,

сменить диск 1,

выполнить повторную зеркализацию с помощью утилиты INSTALL.NLM (в NetWare 5.x – NWCONFIG.NLM).

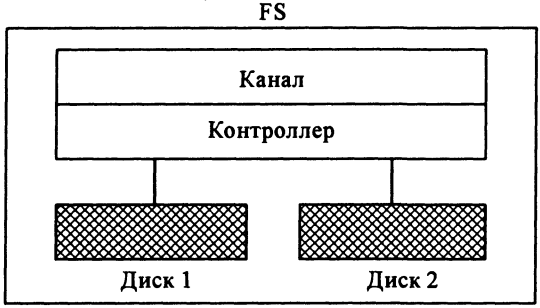


Рис. 8.37. Зеркальное отображение дисков

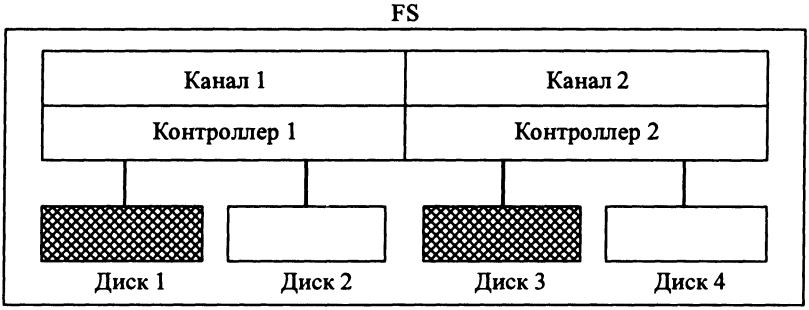


Рис. 8.38. Дублирование дисков

Дублирование дисков. Современный интерфейс IDE включает два канала, к каждому из которых можно подключить два диска (рис. 8.38). Если выполняется зеркализация дисков, которые подключены к разным каналам (диски 1, 3), то говорят о дублировании дисков. В этом случае операции вывода на диски 1 и 3 ОС выполняет параллельно. При наличии очереди запросов на чтение данных с диска 1 ОС NetWare организует параллельное чтение блоков с диска 1 и диска 3.

Система отслеживания транзакций TTS. NetWare автоматически отслеживает транзакции, связанные с системной базой данных сетевых ресурсов (Bindery – NetWare 3.x, NDS – NetWare 4.x/5.x) и предоставляет API-интерфейс для ведения транзакций прикладными программами.

Системная база данных сетевых ресурсов (СБДСР) включает следующие взаимосвязанные файлы: файл объектов, файл атрибутов и файл значений этих атрибутов.

Пусть A, B, C – блоки соответствующих файлов, где хранится информация о пользователе USER, и *a*, *b*, *c* – записи о пользователе USER в этих блоках. Предположим, что администратор удаляет пользователя USER из СБДСР, т. е. данные *c*, *в*, *a* должны быть исключены из блоков C, B, A. При выполнении этих изменений система TTS записывает в транзакционный файл BACKOUT.TTS (том SYS) данные *c*, *в*, *a* до изменения соответствующих блоков (верхняя стрелка на рис. 8.39).

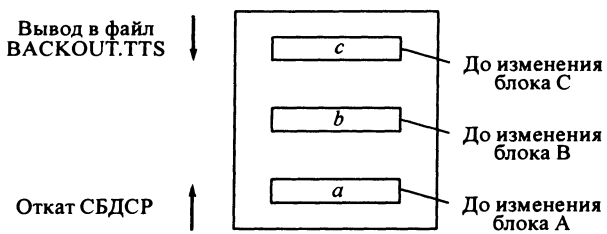


Рис. 8.39. Транзакционный файл BACKOUT.TTS

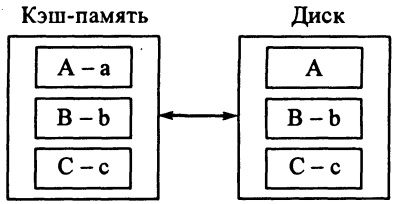


Рис. 8.40. Возможное состояние кэш-памяти и диска

Допустим, что при перезаписи данных из ОП на диск на файловом сервере произошел сбой (т. е. он «завис»). На рис. 8.40 показаны возможные состояния кэш-памяти и диска. Знак «–» на рисунке означает, что блок не содержит соответствующие записи. Из рис. 8.40 видно, что в силу случайности блок А после изменения (А–а) не был перезаписан из кэша на диск. При перезапуске файлового сервера ОС NetWare открывает файл BACKOUT.TTS и выполняет откат СБДСР, т. е. система TTS читает с конца файла записи до изменений (стрелка внизу на рис. 8.39) и восстанавливает их в СБДСР. В результате блоки С, В, А будут восстановлены (рис. 8.41, *а*). Необходимо отметить, что данные одной транзакции будут потеряны, но будет обеспечена целостность СБДСР, что намного важнее.

Если ОС NetWare не ведет транзакции (TTS можно отключить с помощью консольной команды DISABLE TTS), то после восстановления системы СБДСР будет содержать данные, соответствующие состоянию диска перед сбоем (см. рис. 8.40 и 8.41, *б*), т. е. в этом случае произойдет нарушение целостности БД, пользователь не удален из СБДСР, но его свойства b и их значения с оказались потерянными.

Использование устройств бесперебойного питания (UPS). Как отмечалось выше, сбой в системе, в частности внезапное отключение питания, может привести к потере данных. Для защиты файлового сервера от скачков напряжения в электросети используют специальные устройства бесперебойного питания UPS (Unterruptible Power Supply) (рис. 8.42).

С TTS	Без TTS
Операции, выполняемые с системной базой данных сетевых ресурсов	
C – c	C – c
B – b	B – b
A – a	A – a
Сбой	
Состояние СБДСР после восстановления системы	
A	A
B	B – b
C	C – c
а	*б*

Рис. 8.41. Работа системы с TTS (*а*) и без TTS (*б*)

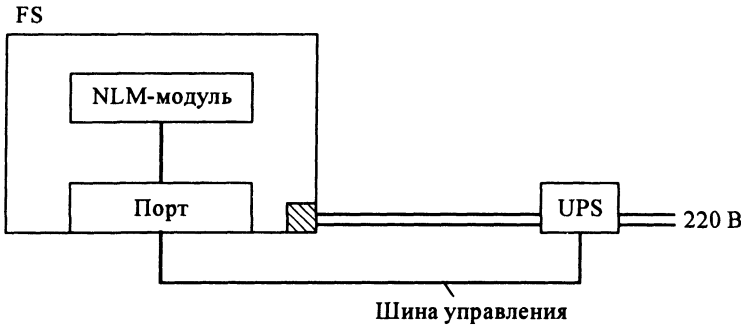


Рис. 8.42. Взаимодействие файлового сервера с устройством бесперебойного питания

Предположим, что входное напряжение электросети упало ниже некоторого порога. В этом случае UPS подает напряжение на вход файлового сервера от своей внутренней батареи. Время, в течение которого батарея разряжается и поддерживает требуемое напряжение на файловом сервере, зависит от типа UPS и для простого устройства (например BACK-UPS) составляет 5...7 мин. Одновременно UPS передает сигнал на порт файлового сервера по шине управления. Этот сигнал вырабатывает прерывание, которое обрабатывает специальный NLM-модуль, который переходит в состояние ожидания на время, которое было указано при запуске этого модуля (как правило, оно равно времени работы батареи). Если в течение этого времени ожидания напряжение электросети не восстанавливается, то после активизации NLM-модуль выполняет команду DOWN. При этом сервер нормально завершает работу: все «грязные» блоки кэш-памяти перезаписываются на диск, все открытые файлы закрываются и пользователи оповещаются об останове файлового сервера. Если за время ожидания напряжение электросети восстанавливается, то по шине управления на порт файлового сервера поступает сигнал, который завершает выполнение NLM-модуля, связанного с этим портом. Далее работа сервера продолжается в прежнем режиме.

Существуют различные средства поддержки UPS-мониторинга файлового сервера NetWare. В продукт PowerChute Plus v.4.0.1 for NetWare включены средства (NLM-модули, кабели управления и т.д.), позволяющие обеспечить UPS-мониторинг через COM-порт. В состав NetWare входит модуль UPS.NLM, поддерживающий UPS-мониторинг через один из портов:

- DCB- или HBA-адаптера,
- специальной карты UPS MONITORING BOARD.

Третий уровень надежности SFT-III

Третий уровень надежности SFT-III обеспечивается использованием ОС NetWare SFT III v.3.x, 4.x, 5.x. Эта ОС управляет работой двух файловых серверов. Один из них функционирует в режиме «горячего» резервирования (рис. 8.43).

Рис. 8.43. Зеркальное отражение файловых серверов

В режиме нормального функционирования основной сервер принимает и обрабатывает кадры, передаваемые по шине. При этом по специальному кабелю MSL (его длина не превышает 2 км) на резервный сервер передаются все изменения, связанные с внешней и основной памятью. Резервный сервер посылает также по кабелю MSL специальные сообщения с целью определения состояния основного сервера: работоспособен он или нет. Эти сообщения передаются несколько раз в секунду.

Когда резервный сервер обнаружит, что основной сервер не работоспособен (например, в результате «зависания»), то он начинает принимать и обрабатывать кадры, передаваемые по шине. Основной сервер можно отремонтировать или заменить на новый. Как только резервный сервер обнаружит, что основной сервер перешел в работоспособное состояние, то он активизирует процесс выравнивания внешней и основной памяти этого главного сервера. После этого основной сервер возобновляет обработку кадров сети, а второй сервер переходит в состояние «горячего» резервирования.

Поддержка дисковых массивов RAID

Когда объем внешней памяти файлового сервера приближается к 10 Гб, то использование традиционных способов обеспечения надежного хранения данных (см. SFT II) становится проблематичным. В настоящее время для надежного хранения больших объемов данных (порядка 10 и более Гб) используют дисковые массивы RAID (Redundant Array of Independent Disks), которые представляют собой специальные устройства, подключаемые к файловому серверу по SCSI-интерфейсу. Как правило, в корпусе устройства RAID устанавливают более 5 дисков (рис. 8.44). Объем каждого диска зависит от типа устройства и колеблется от 1 до 4 Гб. Существует несколько систем RAID: Digital Storage Works RAID Array 210, HP Disk System, Micropolis RAIDion LTX, Storage Dimensions SuperFlex и др. Все они поставляются с требуемым

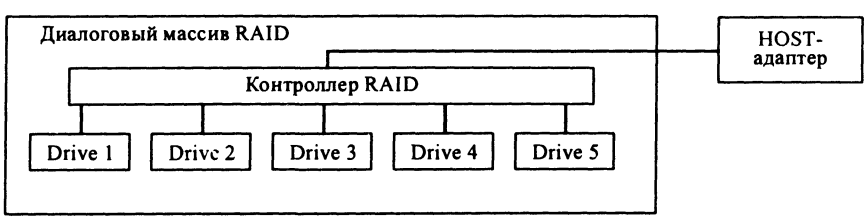


Рис. 8.44. Дисковый массив RAID

RAID Level	Drive 1	Drive 2	Drive 3	Drive 4	Drive 5
1	Sector 1 Sector 2 ... Sector 20	Sector 1 Sector 2 ... Sector 20			
3 Sector 1 Byte Sector 2 Byte ... Sector 20 Byte	1,5, ..., 509 1,5, ..., 509 1,5, ... , 509	2,6, ..., 510 2,6, ..., 510 2,6, ..., 510	3,7, ..., 511 3,7, ..., 511 3,7, ..., 511	4,8, ..., 512 4,8, ..., 512 4,8, ..., 512	Parity Parity Parity
4	Sector 1 Sector 5 Sector 9 Sector 13 Sector 17	Sector 2 Sector 6 Sector 10 Sector 14 Sector 18	Sector 3 Sector 7 Sector 11 Sector 15 Sector 19	Sector 4 Sector 8 Sector 12 Sector 16 Sector 20	Parity Parity Parity Parity Parity
5	Sector 1 Sector 5 Sector 9 Sector 13 Parity	Sector 2 Sector 6 Sector 10 Parity Sector 17	Sector 3 Sector 7 Parity Sector 14 Sector 18	Sector 4 Parity Sector 11 Sector 15 Sector 19	Parity Sector 8 Sector 12 Sector 16 Sector 20

Рис. 8.45. Уровни RAID

набором NLM-модулей и поэтому поддерживаются NetWare. Дисковый массив настраивается на определенный уровень RAID (рис. 8.45). Разработаны стандарты на уровни 0 – 5. Другие уровни RAID (6, 7, 10), используемые в дисковых массивах, не стандартизованы и представляют собой комбинации или модификации уровней 0 –5.

Рассмотрим уровни RAID подробнее. На рис. 8.45 для соответствующих уровней RAID представлены схемы размещения 20 секторов (блоков) какого-либо файла.

Уровень 0. Данные размещаются на нескольких дисках, воспринимаемых компьютером как одно устройство хранения информации большой емкости. Возможно одновременное проведение нескольких операций чтения или записи на разных дисках.

При отказе одного диска данные не восстанавливаются и вся система выходит из строя.

Уровень 1 (см. рис. 8.45). Зеркальное отражение дисков. Дублирование данных обеспечивает высокую отказоустойчивость. Эта схема хранения данных не экономична, так как для каждого диска с данными требуется резервный диск.

Уровень 2. Поочередное размещение битов по дискам. Используется очень редко из-за сложности корректировки ошибок.

Уровень 3 (см. рис. 8.45). Байты сектора поочередно размещаются на нескольких дисках. Сектор как-бы «размазывается» по четырем дискам. Один диск отводится для хранения контрольной информации.

Если часть сектора не читается с какого-либо диска, то система читает все остальные части сектора и контрольную информацию Parity для этого сектора, а затем используя эти данные, восстанавливает недостающую часть сектора.

В операциях чтения и записи сектора участвуют все диски массива, поэтому невозможно параллельное выполнение нескольких операций.

Уровень 4 (см. рис. 8.45). Секторы данных поочередно размещаются на нескольких дисках. Один диск отводится для хранения контрольной информации.

Если, например, не читается сектор 2, то система читает секторы 1, 3, 4, контрольную информацию Parity для этих секторов, а затем, используя эти данные, восстанавливает сектор 2. Возможно параллельное чтение секторов, расположенных на разных дисках.

При записи данных на диск обновляется и контрольная информация. Так как эта информация располагается на одном диске, то невозможно параллельное выполнение операций чтения и обновления контрольных данных.

Уровень 5 (см. рис. 8.45). Секторы данных, а также контрольная информация поочередно размещаются на нескольких дисках. Этот уровень хранения данных используется наиболее часто.

Если, например, не читается сектор 2, то система RAID читает секторы 1, 3, 4, контрольную информацию для этих секторов и, используя эти данные, восстанавливает сектор 2.

Возможно параллельное чтение и запись секторов, расположенных на разных дисках. Например, при записи секторов 3 и 6 эти операции могут выполняться параллельно, так как эти секторы и их контрольная информация располагаются на разных дисках.

Поддержка многопроцессорности

Известно, что суперсерверы (многопроцессорные компьютеры) поддерживают два режима функционирования:

симметричную многопроцессорную обработку (SMP – Symmetric Multi-Processing),

асимметричную многопроцессорную обработку (ASMP – ASymmetric MultiProcessing).

В настоящее время SMP-режим работы суперсервера обеспечивают сетевые ОС Windows NT, UNIX. ОС NetWare 4.1 SMP также поддерживает SMP-режим. Из всего семейства ОС NetWare только NetWare SFT III, использующая зеркальное отображение серверов, поддерживает режим ASMP: в двухпроцессорном сервере первый процессор занимается предоставлением услуг, а второй – операциями ввода/вывода.

Обычно применение NetWare SFT III вызывает снижение производительности до 30 % однопроцессорного сервера, но использование двухпроцессорной конфигурации позволяет серверу SFT III достичь 90 % производительности обычного сервера NetWare с сохранением всех преимуществ зеркального отражения серверов.

Для повышения производительности сетевых ОС используется распределенная параллельная обработка – DPP (Distributed Parallel Processing), состоящая из трех этапов.

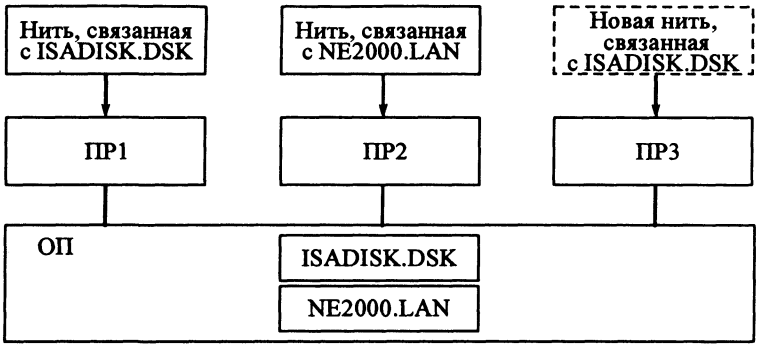


Рис. 8.46. Организация выполнения нитей по SMP-технологии

1. Реализация на серверах NetWare технологии SMP. Для технологии *S*MP (NetWare 4.1 SMP) характерны следующие особенности (рис. 8.46):

- все процессоры имеют общее поле основной памяти;
- если возможно, то новой нити предоставляется свободный процессор.

Предположим, что по запросу от РС образуется новая нить, связанная с драйвером жесткого диска ISADISK.DSK, и если процессор ПР3 свободен, то она будет выполняться на этом процессоре.

NetWare 5.x имеет *мультипроцессорное ядро SMP* с защитой памяти и вытесняющей многозадачностью. В процессе инсталляции система читает из BIOS таблицу конфигурации процессоров и включает в конфигурационный файл STARTUP.NCF требуемые модули Platform Support Modules (PSM), имеющие расширение *.PSM. В этот же файл помещается команда SET Auto Start Processors = ON, позволяющая при старте системы автоматически активизировать *вторичные процессоры* сервера. *Первичный процессор* (processor 0) – это процессор, на котором выполняется управляющая программа NetWare. При запуске процесса (по команде LOAD) система определяет наименее загруженный процессор и ставит процесс к нему в очередь. Если в дальнейшем окажется, что процессор перегружен, то NetWare автоматически перераспределит часть выполняющихся на нем процессов (нитей) на менее загруженный процессор.

2. Дальнейшее расширение доменной архитектуры (см. § 8.1). В Net Ware 5.x используется доменная защита, но, в отличие от NetWare 4.1, число доменов не ограничено.

3. Поддержка технологии ASMP и групп серверов. Для технологии ASMP характерны следующие особенности (рис. 8.47):

- с каждым процессором связана своя область основной памяти;
- каждый NLM-модуль загружается в одну основную память какого-нибудь процессора, т. е. NLM-модуль закрепляется за процессором.

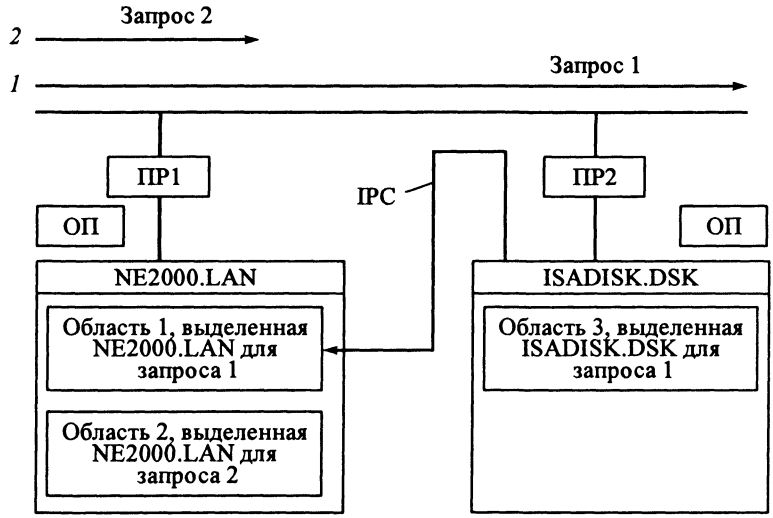


Рис. 8.47. Организация выполнения запросов по ASMP-технологии

Рассмотрим пример. Предположим, что модуль NE2000.LAN закреплен за процессором ПР1, а модуль ISADISK.DSK – за процессором ПР2. Запрос 1, поступающий с PC, сначала обрабатывается NLM-модулем NE2000.LAN, который при обработке этого запроса выделяет область 1 в основной памяти процессора ПР1. Затем запрос 1 обрабатывается драйвером жесткого диска ISADISK.DSK, который выполняется на другом процессоре ПР2. NLM-модуль ISADISK.DSK обращается к области 1 памяти процессора ПР1, посылая запросы через межпроцессорную линию связи IPC (Inter – Processor Communications). В данном примере процессоры ПР1 и ПР2 образуют своеобразный конвейер, используемый для обработки запросов от рабочих станций.

Ключевым элементом технологии ASMP фирмы Novell является менеджер блокировок Lock Manager. Это программное обеспечение, разработанное компанией NetFRAME Systems на основе технологии корпорации Oracle, позволяет множеству процессоров совместно работать над одним запросом. Менеджер блокировок координирует запросы IPC и контролирует доступ к данным так, чтобы блок данных одновременно обновлялся не более чем одним процессором.

Помимо поддержки ASMP на этом этапе происходит распространение доменной архитектуры за пределы единичного сервера. В один домен будут входить процессоры нескольких серверов. Это достигается благодаря использованию распределенной файловой системы, которая позволяет серверам сети обмениваться запросами и ответами.

Группа серверов, будь они однопроцессорные или многопроцессорные, может работать как одна система и собирать воедино свои свободные ресурсы. Эти кластеры серверов могут использовать высокоскоростные линии связи, чтобы передать задание незагруженным процессорам домена. В каком-то смысле эти серверы образуют из процессоров сети виртуальный процессор.

8.6. Механизмы защиты информации

Авторизация доступа к данным сети

В ОС NetWare реализованы три уровня защиты данных (рис. 8.48). Под аутентификацией здесь понимается

- процесс подтверждения подлинности клиента при его подключении к сети;
- процесс установления подлинности пакетов, передаваемых между сервером и РС.

Права по отношению к файлу (каталогу) определяют, какие операции пользователь может выполнить с файлом (каталогом). Администратор для каждого клиента сети определяет права по отношению к любому сетевому файлу или каталогу.

Атрибуты определяют некоторые системные свойства файлов (каталогов). Они могут быть назначены администратором для любого сетевого файла или каталога.

Например, чтобы записать данные в файл, клиент должен:

знать свой идентификатор и пароль для подключения к сети;

иметь право записи данных в этот файл;

файл должен иметь атрибут, разрешающий запись данных.

Атрибуты файла (каталога) имеют более высокий приоритет, чем права пользователей к этому файлу.

Аутентификация пользователей при подключении к сети

Подключение к сети выполняется с помощью утилиты LOGIN.EXE. Эта программа передает на сервер идентификатор, введенный пользователем (рис. 8.49). По этому идентификатору NetWare выполняет поиск соответствующего объекта пользователя в системной БД сетевых ресурсов. Если в БД хранится значение пароля для этого клиента, то NetWare посылает на РС зашифрованный

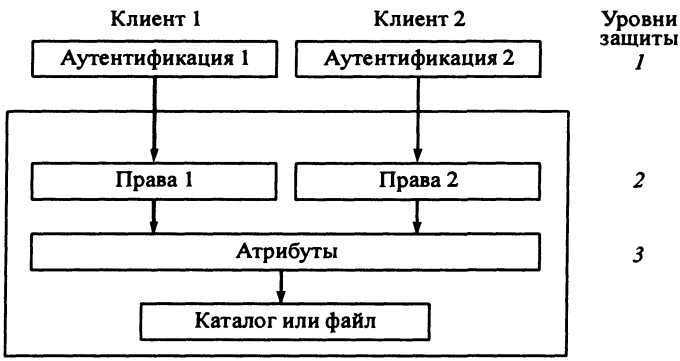


Рис. 8.48. Уровни защиты данных в ОС NetWare

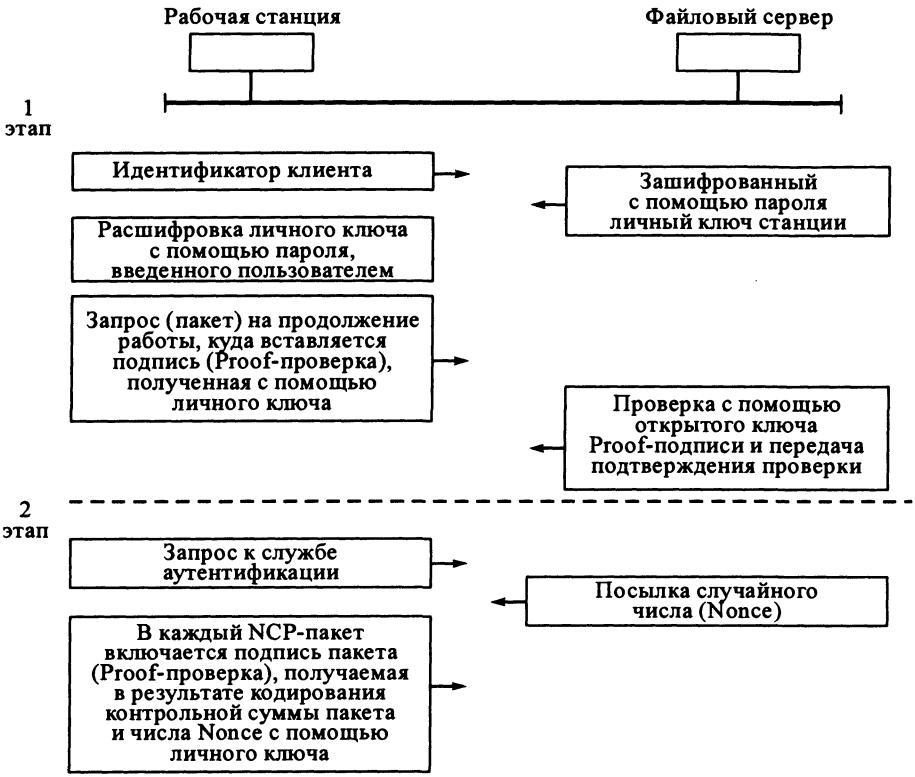


Рис. 8.49. Аутентификация клиента

с помощью пароля личный (закрытый) ключ станции (симметричное шифрование). На РС этот ключ расшифровывается с помощью пароля, введенного пользователем, и используется для получения подписи запроса (пакета) к серверу о продолжении работы. Сервер расшифровывает эту подпись с помощью открытого ключа (асимметричное шифрование), проверяет ее и посылает подтверждение на РС. В дальнейшем каждый NCP-пакет может снабжаться подписью, получаемой в результате кодирования личным ключом контрольной суммы пакета и случайного числа Nonce. Это число генерируется для каждого сеанса. Поэтому подписи пакетов не повторяются для разных сеансов, даже если пользователь выполняет те же самые действия.

Использование сигнатур для передачи NCP-пакетов

Необходимость применения сигнатуры (подписи) NCP-пакетов связана со скандалом, разыгравшимся в 1992 г. Тогда голландский студент предложил простой способ «взламывания» файлового сервера NetWare, основывавный на параллельной работе хаккера и пользователя, имеющего требуемые права (рис. 8.50).

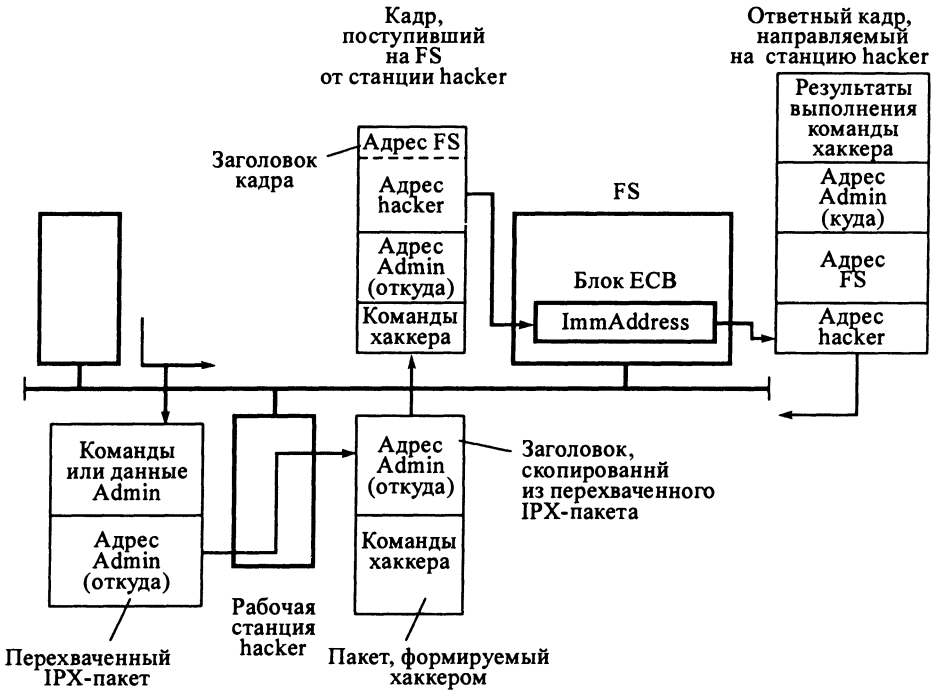


Рис. 8.50. Организация несанкционированного доступа к файловому серверу

На PC хаккера (hacker) функционирует программа, которая перехватывает пакеты, передаваемые по шине сети. При формировании пакета программа хаккера выполняет следующие действия:

- переписывает в заголовок формируемого IPX-пакета заголовок перехваченного пакета;
- записывает в поле данных требуемую команду.

Далее пакет посылается на файловый сервер. Файловый сервер пересылает адрес станции hacker в поле ImmAddress блока ECB и использует данные заголовка пакета IPX, чтобы определить номер соединения и возможность выполнения команды. Но в заголовке пакета хаккера записан адрес пользователя (адрес Admin), который имеет требуемые права. Поэтому команда хаккера выполняется.

При формировании сетевым адаптером заголовка ответного кадра адрес станции, куда непосредственно передается кадр, выбирается из поля ImmAddress блока ECB, т. е. станция hacker воспринимается файловым сервером как маршрутизатор или мост. Напомним, что адрес конечной станции-получателя хранится в заголовке пакета IPX (в данном случае это адрес Admin, хотя для хаккера это не имеет значения). Таким образом, ответ посылается на станцию hacker, где и обрабатывается.

Подпись NCP-пакета (специальное поле в этом пакете) делает невозможным параллельную работу хаккера и пользователя. Подпись (сигнатура) пакета – это шифр, для формирования которого используется номер пакета, его содержимое и случайное число Nonce. Шифр создается с помощью открытого ключа. Важно отметить, что сигнатура изменяется в каждом пакете. Спрогнозировать последовательность подписей практически невозможно.

NCP-пакеты могут подписываться и PC, и файловым сервером. Для инициирования включения подписи в NCP-пакеты администратор должен выполнить следующие действия (для NetWare 3.12/ 4.x/5.x):

1. С консоли файлового сервера необходимо ввести SET-команду

SET NCP Packet Signature Option = уровень (по умолчанию 1).

Здесь уровень имеет одно из следующих значений:

0 – сервер не подписывает пакет,

1 – сервер подписывает пакет, если этого требует клиент (уровень на станции больше или равен двум),

2 – сервер подписывает пакет, если клиент также способен это сделать (уровень на станции больше или равен 1),

3 – сервер подписывает пакет и требует этого от всех клиентов (иначе подключение к сети невозможно).

2. На PC в конфигурационный файл (например, в раздел Netware DOS Reques-ter файла net.cfg) необходимо включить строку:

Signature Level = уровень (по умолчанию 1)

Можно задать один из следующих уровней:

0 – клиент не подписывает пакет,

1 – клиент подписывает пакет, если этого требует сервер (уровень на сервере больше или равен 2),

2 – клиент подписывает пакет, если сервер также способен это сделать (уровень на сервере больше или равен 1),

3 – клиент подписывает пакет и требует этого от всех серверов (иначе подключение к сети невозможно).

В табл. 8.14 перечислены различные сочетания уровней на сервере и PC, а также варианты подписи пакета.

Таблица 8.14. **Варианты подписи пакета**

Уровень на PC	Уровень на сервере			
	0	1	2	3
0	–	–	–	N
1	–	–	+	+
2	–	+	+	+
3	N	+	+	+

Примечание: + – пакеты подписываются; – – пакеты не подписываются; N – PC не подключается к сети.

Определение эффективных прав пользователей к каталогам и файлам

Права, которые могут быть предоставлены пользователю (или группе пользователей) по отношению к каталогу или файлу, перечислены в табл. 8.15.

Таблица 8.15. **Список возможных прав к каталогу или файлу**

Право	Обозначение	Описание
Supervisor	S	Предоставляет все права по отношению к каталогу или файлу, включая возможность назначения этого права другим пользователям. Не блокируется фильтром наследуемых прав IRF. Это право не может быть удалено ниже по дереву каталогов
Read	R	Чтение существующего файла (просмотр содержимого текстового файла, просмотр записей в файле БД и т. д.)
Write	W	Запись в существующий файл (добавление, удаление частей текста, редактирование записей БД)
Create	C	1. Создание в каталоге новых файлов (и запись в них) и подкаталогов. 2. На уровне файла позволяет восстанавливать файл, если он был ошибочно удален
Erase	E	Удаление существующих файлов и каталогов
Modify	M	Изменение имен и атрибутов (файлов и каталогов), но не содержимого файлов
File Scan	F	1. Просмотр в каталоге имен файлов и подкаталогов. 2. По отношению к файлу – возможность видеть структуру каталогов от корневого уровня до этого файла (путь доступа)
Access Control	A	Возможность предоставлять другим пользователям все права, кроме Supervisor. Возможность изменять фильтр наследуемых прав IRF

Права и фильтры (маски) наследуемых прав назначаются администратором сети с помощью утилит NetWare. Но назначение прав для каждого пользователя по отношению ко всем требуемым файлам и каталогам является утомительной задачей. В NetWare предлагается механизм наследования прав. Введем некоторые определения.

- *опекун* (Trustees) – это пользователь (или группа пользователей, или другой объект), которому администратор с помощью утилиты (например, FILER) явно назначает права к какому-либо файлу или каталогу. Такие права называются опекунскими назначениями;
- *фильтр наследуемых прав* (IRF – Inherited Right Filter) – свойство файла (каталога), определяющее, какие права данный файл (каталог) может унаследовать от родительского каталога. Фильтр назначается администратором с помощью утилиты (например FILER);

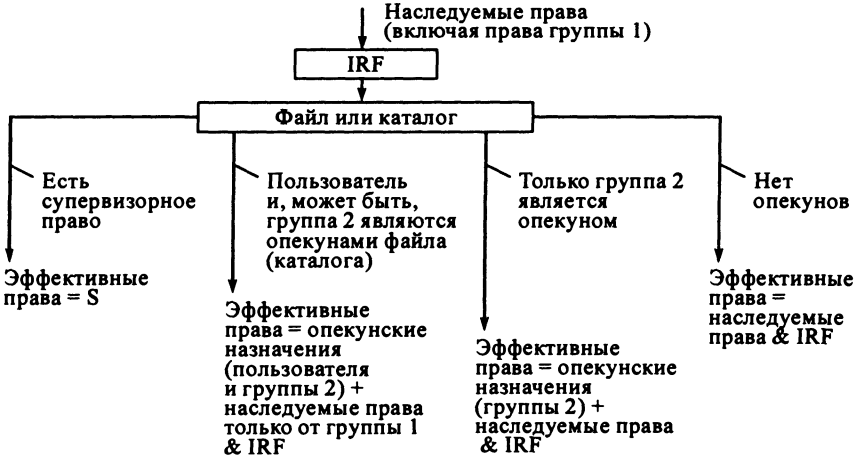


Рис. 8.51. Схема определения ОС NetWare эффективных прав пользователя к файлу или каталогу:
+ и & – логические операции ИЛИ и И

• *наследуемые права* – права, передаваемые (распространяемые) от родительского каталога;

• *эффективные права* – права, которыми пользователь реально обладает по отношению к файлу или каталогу.

На рис. 8.51 представлена схема формирования ОС NetWare эффективных прав пользователя к файлу или каталогу. Здесь предполагается, что пользователь является участником групп 1 и 2.

Важно подчеркнуть, что к этим эффективным правам добавляются эффективные права тех пользователей, которые указаны в списке Security Equal To (эквивалентны по защите).

Рассмотрим пример, иллюстрирующий приведенную выше схему. Предположим, что пользователь John работает с каталогами D1, D2, D3 и файлом F1:

FS/VOL: D1\D2\D3\F1

и не является участником групп. В табл. 8.16 показаны изменения эффективных прав пользователя John по отношению к указанным каталогам и файлу.

Таблица 8.16. **Изменение эффективных прав пользователя**

Право	Права и фильтры	Примечания
Корень тома FS/VOL:		
Наследуемые права	–	
IRF	SRWCEMFA	
Опекунские права для John	–	
Права пользователей из списка Security Equal To	–	
Эффективные права	–	

Окончание табл. 8.16

Право	Права и фильтры	Примечания
	Каталог D1	
Наследуемые права	–	
IRF	SRWCEMFA	
Опекунские права для John	RW F	
Права пользователей из списка Security Equal To	–	
Эффективные права	RW F	См. вторую ветвь на рис. 8.51
	Каталог D2	
Наследуемые права	RW F	
IRF	SR CEMFA	
Опекунские права для John	–	
Права пользователей из списка Security Equal To	A	
Эффективные права	R FA	См. четвертую ветвь на рис. 8.51 + права от пользователей, эквивалентных по защите
	Каталог D3	
Наследуемые права	R FA	
IRF	S WCEM	
Опекунские права для John	S	
Права пользователей из списка Security Equal To	A	
Эффективные права	SRWCEMFA	См. первую ветвь на рис. 8.51
	Файл F1	
Наследуемые права	SRWCEMFA	
IRF	FA	
Опекунские права для John	WC	
Права пользователей из списка Security Equal To	A	
Эффективные права	SRWCEMFA	См. первую ветвь на рис. 8.51 (право S не маскируется)

Атрибуты каталогов и файлов

Атрибуты файла (каталога) устанавливаются администратором сети с помощью утилиты (например FLAG) и управляют доступом к этому файлу или каталогу. Атрибуты файлов и каталогов перечислены в табл. 8.17.

Таблица 8.17. **Атрибуты файлов и каталогов NetWare**

Атрибут	Обозначение	Описание
Атрибуты NetWare 3.x/4.x/5.x		
Delete Inhibit	Di	Запрещает пользователю удалять файл или каталог
Hidden	H	Делает файл или каталог невидимым для команды DIR и предотвращает его копирование и удаление. Однако команда NDIR позволяет его увидеть, если пользователь обладает правом File Scan для каталога
Purge	P	Указывает ОС физически затирать файл или каталог при его удалении (delete). После этого файл или каталог нельзя восстановить
Rename Inhibit	Ri	Запрещает пользователю переименовывать файл или каталог
System	Sy	Устанавливается для файлов или каталогов, используемых только ОС (далее см. атрибут H)
Normal	N	Опция (не атрибут) утилиты FLAG, позволяющая сбросить все атрибуты
All	All	Опция (не атрибут) утилиты FLAG, позволяющая установить все атрибуты
Следующие атрибуты справедливы только для файлов		
Archive Needed	A	Автоматически устанавливается для файлов, которые были модифицированы, но никуда не архивировались
Copy Inhibit	Ci	Запрещает копировать файл (только для ОС Macintosh)
Execute Only	X	Защищает файл от копирования. Устанавливается для файлов *.EXE и *.COM. Только пользователь с правами Supervisor может установить этот атрибут. Этот атрибут не может быть снят даже пользователем с правами Supervisor. Файл с этим атрибутом можно только удалить
Read Only	Ro	Запрещает запись в файл, автоматически устанавливаются атрибуты Rename Inhibit и Delete Inhibit. NetWare показывает значение Read Write (Rw), когда атрибут Read Only снимается
Shareable	Sh	Позволяет нескольким пользователям одновременно получать доступ к файлу (обычно используется с Ro)
Transactional	T	Показывает, что TTS ведет неявные транзакции для этого файла

Окончание табл. 8.17

Атрибут	Обозначение	Описание
Атрибуты только для NetWare 4.x/5.x		
Don't Migrate	Dm	Запрещает миграцию (перемещение) файла или каталога на магнитооптическое устройство
Immediate Compress	Ic	Для файла – файл будет сжат как только процессор освободится. Для каталога – файлы сжимаются после их изменения или при копировании файлов в каталог
Don't Compress	Dc	Запретить системе сжимать файл или каталог
Флаги статуса файла		
Compressed	Co	Показывает, что файл сжат
Can't Compress	Cc	Показывает, что файл не может быть сжат
Migrated	M	Файл был перезаписан на магнитооптический диск

Права доступа к объектам NDS и их свойствам

Выше отмечалось, что СБДСР представляет собой совокупность объектов, их свойств и значений этих свойств. В NetWare 4.x/5.x эта БД называется NDS (NetWare Directory Services), а в NetWare 3.x – Bindery. Отличия NDS от Bindery описаны в отдельном разделе, который посвящен глобальному сетевому каталогу (NDS).

Объекты NDS связаны между собой в иерархическую структуру, которую часто называют деревом NDS. На верхних уровнях дерева (ближе к корню [Root]) описываются логические ресурсы, которые принято называть контейнерными объектами. На самом нижнем (листьевом) уровне располагаются описания физических ресурсов, которые называют оконечными объектами.

В качестве контейнерных объектов используются объекты типа [Root] (корень), C (страна), O (организация), OU (организационная единица). Оконечные объекты – это User (пользователь), Group (группа), NetWare Server (сервер NetWare), Volume (том файлового сервера), Directories (директория тома) и т. д. Оконечные объекты имеют единое обозначение – CN.

В NetWare 4.x/5.x разработан механизм защиты дерева NDS. Этот механизм очень похож на механизм защиты файловой системы, который был рассмотрен выше. Чтобы облегчить понимание этого механизма, оконечный объект можно интерпретировать как файл, а контейнерный объект – как каталог, в котором могут быть созданы другие контейнерные объекты (как бы подкаталоги) и оконечные объекты (как бы файлы). На рис. 8.52 представлена схема дерева NDS.

В отличие от файловой системы в механизме защиты дерева права по отношению к какому-либо объекту можно предоставить любому контейнерному или оконечному объекту дерева NDS. В частности допустимо рекурсивное назначение прав объекта по отношению к этому же объекту.

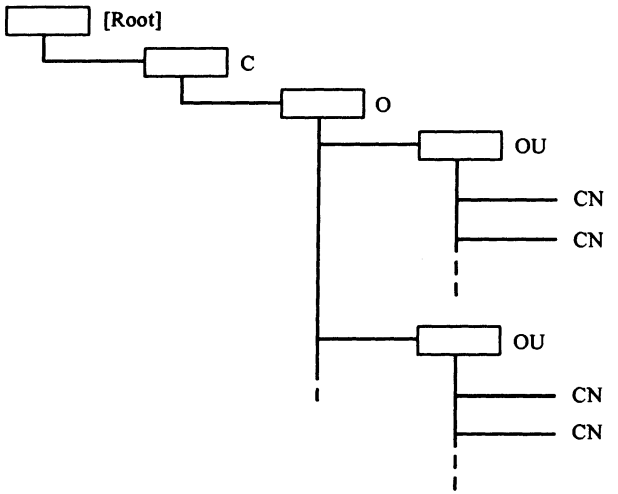


Рис. 8.52. Схема дерева *NDS*:
[Root], C, O, OU – контейнерные объекты; CN – оконечные объекты

Права, которые могут быть предоставлены объекту по отношению к другому или тому же самому объекту, перечислены в табл. 8.18.

Таблица 8.18. **Список возможных прав по отношению к объекту**

Право	Обозначение	Описание
Supervisor	S	Гарантирует все привилегии по отношению к объекту и его свойствам. В отличие от файловой системы это право может быть блокировано фильтром наследуемых прав IRF, который может быть назначен для каждого объекта
Browse	B	Обеспечивает просмотр объекта в дереве NDS
Create	C	Это право может быть назначено только по отношению к контейнерному объекту (контейнеру). Позволяет создавать объекты в данном и во всех дочерних контейнерах.
Delete	D	Позволяет удалять объект из дерева NDS
Rename	R	Позволяет изменять имя объекта

Администратор сети может для каждого объекта в дереве NDS определить значения свойств этого объекта. Для объекта User – это имя Login, требования к паролю, пароль пользователя, пользовательский сценарий подключения и т. д. Механизм защиты NDS предоставляет также возможность назначать права по отношению к свойствам любого объекта (табл. 8.19).

Таблица 8.19. **Список возможных прав по отношению к свойству объекта**

Право	Обозначение	Описание
Supervisor	S	Гарантирует все привилегии по отношению к свойству объекта. Это право может быть блокировано фильтром наследуемых прав IRF, который может быть назначен для свойства. Фильтры IRF назначаются для объекта и его свойства отдельно
Compare	C	Позволяет при поиске объекта (например, с помощью утилиты NLIST) сравнивать значение свойства с любой константой. Однако это право не обеспечивает чтение значения свойства. После операции сравнения возвращается результат: True или False
Read	R	Позволяет читать значение свойства из БД NDS. Право Read включает право Compare
Write	W	Позволяет добавлять, изменять или удалять значение свойства. Право Write включает право Add Self
Add Self	A	Позволяет опекуну (User) добавлять или удалять самого себя как значение свойства. Это право имеет смысл только для свойств, которые содержат имена пользователей в качестве значений, например, для свойства Members (участники) объекта Group (группа).

Права по отношению к свойствам могут быть назначены сразу для всех свойств объекта (All Properties) или для выбранного свойства (Selected Properties). Во втором случае права для выбранного свойства замещают права, назначенные через опцию All Properties. Следует также отметить, что права для выбранного свойства (Selected Properties) не наследуются, а права для всех свойств объекта (All Properties) наследуются.

Права и фильтры наследуемых прав назначаются администратором с помощью утилит NetWare 4.x/5.x (NetWare Administrator). Но назначение прав объектов по отношению ко всем требуемым объектам и свойствам – это утомительная задача. Предлагаемый в NetWare 4.x/5.x механизм наследования прав в дереве NDS напоминает механизм наследования прав в файловой системе.

Определения опекуна (Trustees), фильтра наследуемых прав (IRF), наследуемых прав, эффективных прав совпадают с соответствующими определениями для файловой системы. Только понятия файл, каталог, пользователь (группа пользователей) следует заменить соответственно на оконечный объект, контейнерный объект, произвольный объект. Ниже приведены эти определения.

Рис. 8.53. Схема определения ОС NetWare эффективных прав объекта А к объекту Б (или его свойствам):
+ и & – логические операции ИЛИ и И

Опекун (Trustees) – объект, которому администратор с помощью утилиты (например NetWare Administrator) явно назначает права к какому-либо объекту (или его свойствам). Такие права называются опекунскими назначениями.

Фильтр наследуемых прав (IRF – Inherited Right Filter) – характеристика объекта (или его свойств), определяющая, какие права данный объект (или его свойства) может унаследовать от родительского контейнерного объекта. Фильтр назначается администратором с помощью утилиты (например NetWare Administrator).

Наследуемые права – права, передаваемые (распространяемые) от родительского контейнерного объекта.

Эффективные права – права, которыми пользователь реально обладает по отношению к другому или тому же самому объекту (или его свойствам).

На рис. 8.53 показана схема формирования ОС NetWare эффективных прав объекта А по отношению к какому-либо объекту Б (или его свойствам). Здесь предполагается, что объект А является участником групп 1 и 2. Ясно, что группы следует учитывать только в том случае, если объект А – это объект типа User (пользователь).

Если объект А является пользователем, то к его эффективным правам добавляются эффективные права тех объектов, которые указаны в свойстве Security Equal To этого пользователя.

Рассмотрим пример, иллюстрирующий эту схему. Предположим, что администратор сети Admin создал дерево NDS, представленное на рис. 8.54.

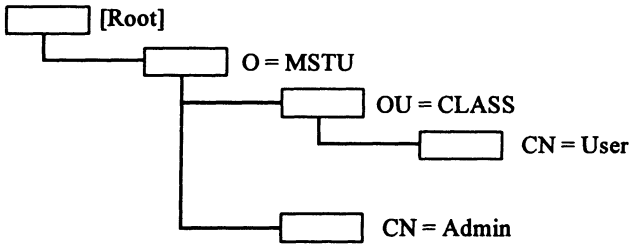


Рис. 8.54. Дерево NDS, которое создает администратор Admin

В табл. 8.20 приведены изменения эффективных прав для пользователей Admin и User по отношению к указанным в таблице объектам (предполагается, что эти пользователи имеют пустые списки Security Equal To).

Таблица 8.20. **Примеры изменения эффективных прав пользователей Admin и User по отношению к объектам**

Право	Пользователь	
	Admin	User
Контейнер [Root]		
Наследуемые права	–	–
IRF	SBCDR	SBCDR
Опекунские назначения	SBCDR	–
Эффективные права	SBCDR	–
Контейнер MSTU		
Наследуемые права	SBCDR	–
IRF	SBCDR	SBCDR
Опекунские назначения	–	–
Эффективные права	SBCDR	–
Контейнер CLASS		
Наследуемые права	SBCDR	–
IRF	–	–
Опекунские назначения	–	SBCDR
Эффективные права	–	SBCDR
Оконечный объект User		
Наследуемые права	–	SBCDR
IRF	SBCDR	SBCDR
Опекунские назначения	–	–
Эффективные права	–	SBCDR

Этот пример иллюстрирует, как администратор Admin может потерять управление контейнерным объектом OU = CLASS. Это может произойти в том случае, если Admin назначает пользователя User опекуном объекта OU = CLASS с правами [SBCDR]. Пользователь User может сбросить все признаки в фильтре наследуемых прав IRF объекта OU = CLASS. Так как право [S] и все

остальные права маскируются, то в этом случае Admin не только потеряет управление объектом OU = CLASS, но он даже не сможет увидеть объект в дереве NDS.

Отметим, что после инсталляции файлового сервера:

- пользователь Admin автоматически получает права [SBCDR] по отношению к объекту [Root];
- фиктивный опекун [Public] автоматически получает право [B] по отношению к объекту [Root]; это означает, что любой пользователь, который подключается к сети, еще до Login видит все объекты дерева NDS (право [B] наследуется от [Root] ко всем объектам).

Санкционирование доступа к консоли файлового сервера

Доступ к консоли файлового сервера можно осуществить посредством основной или удаленной консоли (т. е. с PC).

Администратор может защитить основную консоль, используя пункт меню «Lock File Server Console» утилиты MONITOR.NLM (в NetWare 5.x для этой цели используется утилита сервера SCRSAVER.NLM). После ввода произвольного пароля доступ к консоли (с основной и удаленной) будет заблокирован и администратор может оставить консоль без присмотра. Чтобы разблокировать консоль, администратор должен ввести либо ранее указанный пароль, либо пароль Admin.

Для удаленной консоли необходимо с основной консоли ввести следующие команды:

```
LOAD REMOTE <пароль>
LOAD RSPX
```

Эти команды можно поместить в конфигурационный файл AUTOEXEC.NCF. Затем на PC, где необходимо создать удаленную консоль, следует запустить утилиту SYS:SYSTEM\RCONSOLE.EXE и в диалоге ввести <пароль>, указанный при запуске утилиты REMOTE.

Недостаток такой схемы организации удаленной консоли заключается в том, что параметр <пароль> при загрузке NLM-модуля REMOTE указывается открытым текстом. Чтобы скрыть параметр <пароль>, с основной консоли NetWare 4.x/5.x необходимо ввести следующие команды

```
LOAD REMOVE X
REMOTE ENCRYPT
```

Далее ОС NetWare 4.x/5.x предлагает ввести пароль для удаленной консоли. Он вводится скрытым текстом. В результате ОС формирует <код>, который отображается на консоли файлового сервера.

После этого можно использовать следующие команды:

```
LOAD REMOVE –E <код>
```

(вместо приведенной выше команды можно просто ввести LDREMOTE.NCF)

```
LOAD RSPX
```

На PC указывается пароль, который администратор ввел с основной консоли скрытым текстом.

8.7. Диалоговые интерфейсы

Текстовый и графический интерфейсы ОС

В ОС NetWare используются утилиты следующих типов:

- утилиты командной строки (FLAG, RIGHTS, NDIR и т. д.);
- утилиты-меню (FILER, PCONSOLE и т. д.);
- утилиты, выполняемые под управлением Windows (NetWare Administrator и др.).

Для выполнения утилиты командной строки необходимо ввести имя программы и параметры. Результаты выполнения команды отображаются на экране PC. Как таковой диалог здесь отсутствует.

При работе с утилитой-меню используется текстовый диалоговый интерфейс. Чтобы модифицировать какое-либо поле диалогового окна утилиты-меню, необходимо подсветить это поле и нажать клавишу Enter. Для изменения значения поля на альтернативное используют клавиши «Стрелка влево» и «Стрелка вправо». Если это поле ввода, то появляется курсор и можно отредактировать значение поля. После повторного нажатия клавиши Enter (или клавишей «Стрелка вверх», «Стрелка вниз») подсвечивается следующее поле диалогового окна. Если поле связано со списком, то на экране появляется этот список. Используя клавиши Del и Ins, можно отредактировать этот список значений.

При работе с утилитами, которые выполняются под управлением Windows, используется графический интерфейс, предоставляемый этой оболочкой.

Диалоговый интерфейс пользователя

Иногда требуется, чтобы после загрузки ОС PC неподготовленный пользователь мог бы сразу работать со своими приложениями. Для этого в NetWare предлагаются средства разработки диалогового интерфейса пользователя. Эти средства включают (для NetWare 3.12, 4.x):

- язык описания различных окон меню и их пунктов в текстовых файлах *.SRC;
- утилиту MENUMAKE, транслирующую файл *.SRC в файл *.DAT;
- файл NMENU.BAT для запуска файла *.DAT с описаниями меню пользователя.

На рис. 8.55 представлена структура файла *.SRC. Здесь показана допустимая иерархия операторов, которые можно использовать в файле *.SRC (табл. 8.21).

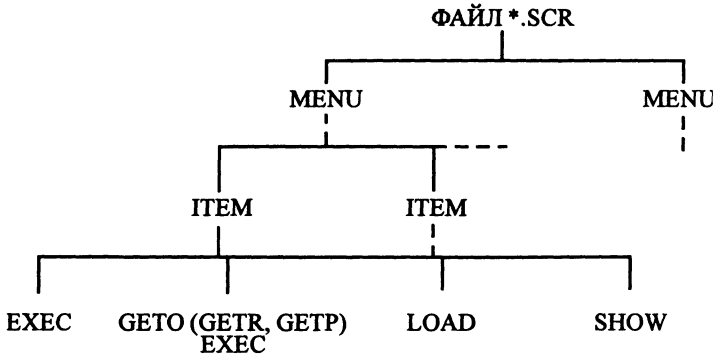


Рис. 8.55. Структура файла *.SCR с описаниями меню

Таблица 8.21. **Операторы, используемые в файле *.SRC**

Оператор	Описание
MENU номер_меню, заголовок_меню	Определяет номер и заголовок меню
ITEM заголовок_пункта {опции}	Определяет заголовок пункта меню
EXEC имя_программы	Определяет имя программы, которая выполняется при выборе пункта меню
SHOW имя_меню	Определяет подменю, которое отображается при выборе пункта меню. Это подменю должно быть описано в том же файле *.SRC
LOAD имя_файла	Определяет подменю, которое отображается при выборе пункта меню. Это подменю должно быть описано в другом файле *.DAT
GETO параметры GETR параметры GETP параметры	Определяют характеристики специального окна, отображаемого на экране для задания параметров программы, которая описывается в следующем операторе EXEC. Операторы GETO, GETR, GETP отличаются способами ввода параметров в специальном окне и их описанием в операторе EXEC (для GETP)

На рис. 8.56 показан пример последовательности операторов в файле *.SRC.

Чтобы предотвратить выход в DOS, команду NMENU, которая запускает диалоговый интерфейс пользователя, часто помещают в файл AUTOEXEC.BAT и используют в цикле.

В NetWare 5.x предлагается интегрированный набор технологий Z.E.N.works Starter Pack, которые обеспечивают следующие возможности

- позволяют управлять PC и рабочими столами пользователей сети;
- предоставляют пользователям сети надежную и простую в использовании сетевую систему, которая поддерживает целостность приложений.

```
MENU                    Определить 1-е меню
     ITEM               Определить 1-й пункт меню
          SHOW          Определить 1-й пункт меню как
                        подменю, определенное в том же
                        файле *.SCR
MENU                    Определить 2-е меню
     ITEM               Определить 1-й пункт меню
          LOAD          Определить 1-й пункт меню как
                        подменю, определенное в другом
                        файле *.DAT
     ITEM               Определить 2-й пункт меню
          EXEC          Определить программу, которая
                        будет выполняться при выборе
                        этого пункта меню
     ITEM               Определить 3-й пункт меню
          GETO          Описать имя поля для задания 1-го
                        параметра программы,
                        определенной ниже в операторе
                        EXEC
          GETO          Описать имя поля для задания 2-го
                        параметра программы,
                        определенной ниже в операторе
                        EXEC
          EXEC          Определить программу, которая
                        будет выполняться при выборе
                        этого пункта меню
```

Рис. 8.56. Пример последовательности операторов в файле *.SRC

Интернационализация диалоговых интерфейсов

ОС NetWare можно настроить так, чтобы сообщения системы отображались на требуемом национальном языке. Можно изменить язык сообщений, отображаемых:

- операционной системой на консоли файлового сервера;
- NLM-модулями на консоли файлового сервера;
- утилитами NetWare на экране PC;
- модулями программного обеспечения клиента на экране PC.

Для изменения языка, используемого для вывода сообщений ОС на консоль файлового сервера, необходимо подготовить соответствующий модуль SERVER.MSG и поместить его в каталог, откуда загружается головная программа ОС NetWare SERVER.EXE.

Для вывода сообщений NLM-модулей на требуемом языке необходимо с консоли файлового сервера ввести команду

LANGUAGE number

где number определяет национальный язык сообщений. В частности число 4 соответствует английскому языку, а 13 – русскому.

На рис. 8.57 представлена структура каталогов, где должны храниться сообщения NLM-модулей. При инсталляции файлового сервера обычно устанавливается только каталог с именем 4, где хранятся модули *.MSG описания сообщений на английском языке для соответствующих NLM-модулей.

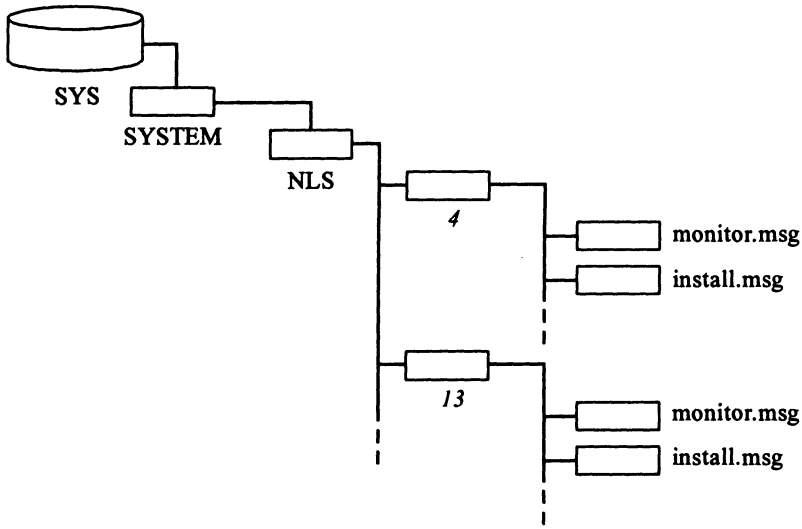


Рис. 8.57. Структура каталогов, где хранятся сообщения NLM-модулей

Для вывода сообщений утилит NetWare и модулей программного обеспечения клиента на требуемом языке используется переменная NWLANGUAGE среды DOS, которая устанавливается, как правило, в файле C:\NWCLIENT\ STARTNET.BAT, создаваемом при инсталляции рабочей станции:

SET NWLANGUAGE=language

где language – один из следующих языков: English, Duetsh, Espanol, Francais, Italiano.

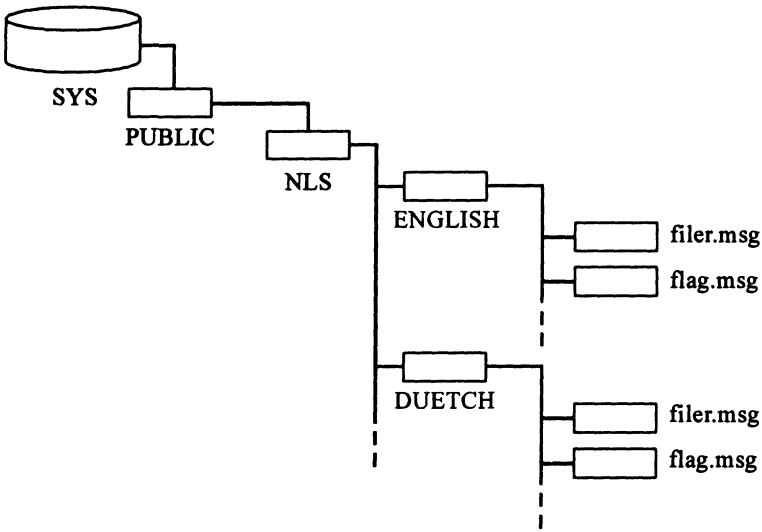


Рис. 8.58. Структура каталогов, где хранятся сообщения утилит NetWare

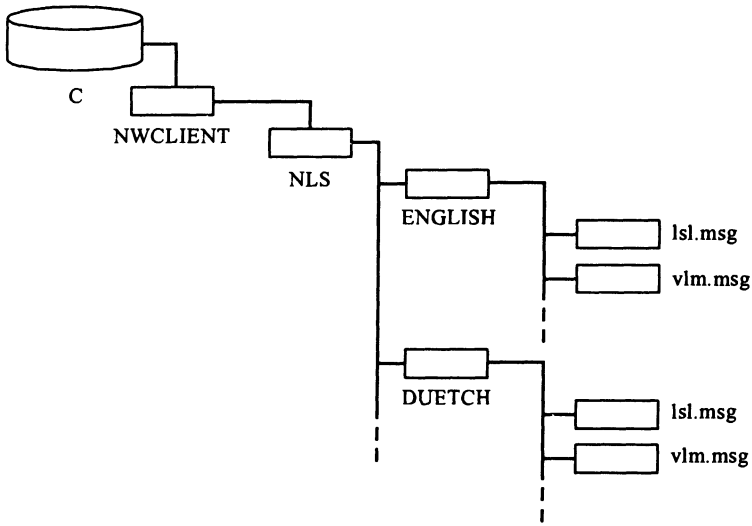


Рис. 8.59. Структура каталогов, где хранятся сообщения модулей программного обеспечения клиента станции

На рис. 8.58 представлена структура каталогов, где должны храниться сообщения утилит NetWare. При инсталляции файлового сервера обычно устанавливается только каталог с именем ENGLISH.

На рис.8.59 изображена структура каталогов и файлов с сообщениями модулей программного обеспечения клиента PC.

При инсталляции PC обычно устанавливается каталог с именем ENGLISH.

Форматы даты, времени и числа, используемых на PC, описываются в файле C:\NWCLIENT\LCONFIG.SYS. Примеры форматов для USA, используемых по умолчанию, представлены в табл. 8.22.

Таблица 8.22. **Примеры форматов для USA**

Страна	Дата	Время	Число
USA	09/22/02	8:37:00 PM	12,345.67

Следует отметить, что в NetWare 4.11 имеется серьезная ошибка, связанная с поддержкой русского языка. При задании на сервере кода страны 7 и кодовой страницы 866 некоторые программы (MHS, FTR Services и др.) работают некорректно. В NetWare 5 осталась проблема поддержки русского языка на сервере.

9. АДМИНИСТРИРОВАНИЕ И ОПЕРАТИВНОЕ УПРАВЛЕНИЕ В ОС NETWARE

Рассмотрены вопросы администрирования операционной и информационной среды. Описана пользовательская операционная среда рабочей станции и файлового сервера. Приведены практические рекомендации по установке и настройке сетевой ОС NetWare 3.x, 4.x/5.x. Изложены основные принципы управления сетевыми ресурсами. Приведен обзор команд оперативного управления и показано, как осуществляется наблюдение и контроль состояния системы. В заключении главы изложены основы разработки приложений для NetWare. Приведены структура и правила написания NLM-модулей и организация API-интерфейса для разработки программ на языке С.

9.1. Администрирование операционной среды

Пользовательская операционная среда рабочей станции

В NetWare в качестве ПО клиента PC используется так называемый 16-разрядный клиент (NetWare 3.x, NetWare 4.x) и 32-разрядный клиент (NetWare 4.x, NetWare 5.x). Рассмотрим особенности установки и организации 16-разрядного клиента (Client16).

Начальная установка ПО 16-разрядного клиента PC выполняется с дискет. Для DOS обычно используют следующие дискеты: NetWare 3.x – WSDOS_1, WSDOS_2, WSDRV_2, NetWare 4.x – WSDOS_1, WSWIN_1, WSDRV_1, WSDRV_2.

Так как NetWare 4.x поставляется на CD-ROM, то дискеты можно создать в DOS, используя утилиту

D:\CLIENT\DOSWIN\MAKEDISK <*диск*>:

где D – драйв CD-ROM, <*диск*> – драйв гибкого диска 3,5″.

Для инсталляции PC необходимо с дискеты WSDOS_1 запустить программу INSTALL.EXE и далее следовать инструкциям этой утилиты.

Для обновления в NetWare 4.x ПО клиента PC можно использовать ту же программу INSTALL.EXE, которая хранится в каталоге SYS:PUBLIC\CLIENT\DOSWIN.

В процессе инсталляции ПО клиента PC, где установлена ОС DOS, на ее локальный диск (обычно C:) записывается каталог NWCLIENT. При этом могут быть также изменены и некоторые стандартные файлы DOS. Пользовательская среда PC настраивается при помощи четырех файлов:

C:\CONFIG.SYS,
C:\AUTOEXEC.BAT,
C:\NWCLIENT\STARTNET.BAT,
C:\NWCLIENT\NET.CFG.

Рассмотрим эти файлы подробнее.

1. CONFIG.SYS

При инсталляции PC в этот файл может быть добавлена команда LASTDRIVE = Z. Эта команда показывает, какие буквы латинского алфавита можно использовать для драйвов.

2. AUTOEXEC.BAT

При инсталляции в этот файл помещается вызов файла STARTNET.BAT, с помощью которого загружаются модули программного обеспечения клиента:

```
@CALL C:\NWCLIENT\STARTNET.BAT
```

Это позволяет автоматизировать подключение клиента к сети при включении компьютера. Данная команда включается в AUTOEXEC.BAT первой строкой. Это не всегда удобно, так как иногда требуется оставить за пользователем право выбора: согласен ли он работать в сети или нет. Для этой цели можно использовать возможности DOS 6.0 и выше по созданию разветвленных файлов загрузки или воспользоваться различными программами типа нортоновской утилиты BE.EXE.

3. STARTNET.BAT

Этот файл создается автоматически при инсталляции рабочей станции и содержит следующие команды:

SET NWLANGUAGE=ENGLISH – переменная среды DOS для вывода сообщений на английском языке,

LSL.COM – программа поддержки связи драйвера сетевого адаптера с протоколами стека,

NE2000.COM – какой-либо драйвер сетевого адаптера,

IPXODI.COM – транспортный протокол SPX/IPX,

VLM.EXE – головная программа запросчика DOS, загружающая файлы *.VLM.

В этом файле не хватает команды регистрации в сети. Поэтому в конец этого файла (или в файл AUTOEXEC.BAT) обычно записывают следующие команды:

```
F:
LOGIN
```

Буква драйва F зависит от конкретной конфигурации PC и означает первый сетевой драйв.

4. NET.CFG

Данный текстовый файл содержит все настройки ПО клиента PC. Он состоит из разделов, каждый из которых включает опции. Заголовок раздела всегда начинается с первой позиции новой строки. Связанные с данным разделом параметры (опции) описываются на следующих строках, но не с первой позиции.

В табл. 9.1 перечислены все возможные разделы, которые можно создать в файле NET.CFG.

Таблица 9.1. **Разделы NET.CFG**

Раздел	Описание
Desktop SNMP	Используется, чтобы поддержать базу данных MIB-II со статистикой о работе сети и коммуникации для протокола SNMP
Link Driver	Определяет конфигурацию драйвера для каждого сетевого адаптера, установленного на PC
Link Support	Используется, чтобы переопределить число и размеры требуемых буферов
Named Pipes	Используется для организации связи PC с приложениями «клиент-сервер» (например с Microsoft SQL Server и т. д.). На PC протокол Named Pipes поддерживается программой DOSNP.EXE
NetBIOS	Используется, чтобы переназначить параметры, используемые протоколом NetBIOS
NetWare DOS Requester	Определяет параметры, используемые запросчиком DOS (файлами VLM.EXE и *.VLM)
NetWare DOS TSA	Используется модулем TSASMS.COM для взаимодействия с SBACKUP.NLM при архивации данных PC (для NetWare 4.x)
Protocol IPX	Определяет параметры протокола IPX
Protocol ODINSUP	Используется, если NDIS-протокол добавлен к стеку протоколов PC. Этот протокол применяется для связи с ОС фирмы IBM: Extended Services и LAN Services
Protocol RFCNBIOS	Используется программой RFCNBIOS.EXE для выполнения команд NetBIOS с использованием протокола TCP/IP
Protocol RPL	Используется, если RPL-протокол добавлен к стеку протоколов PC. Этот протокол применяется для загрузки бездисковых станций (для NetWare 4.x)
Protocol SPX	Определяет параметры протокола SPX

Окончание табл. 9.1

Раздел	Описание
Protocol TCPIP	Используется модулем TCPIP.EXE, если эта программа добавлена в стек протоколов PC
TBMI2	Используется, если необходимо модифицировать среду переключения процессов передачи данных
Transport Provider IPX|UDP	Используется для определения адреса, по которому менеджер SNMP посылает прерывания протокола SNMP

Большинство перечисленных разделов не являются обязательными. При инсталляции PC в NET.CFG автоматически создаются только два раздела (Link Driver и NetWare DOS Requester) с самыми необходимыми опциями (в табл. 9.2 и 9.4 они отмечены слева звездочкой).

В табл. 9.2 перечислены наиболее важные и интересные параметры (опции) разделов Link Driver, Link Support, NetWare DOS Requester и Protocol IPX.

Таблица 9.2. **Опции разделов Link Driver, Link Support, NetWare DOS Requester и Protocol IPX**

Опция	Описание
	Link Driver
*IRQ number	Номер прерывания, на который настроен сетевой адаптер PC
*PORT address	16-ричный адрес порта, на который настроен сетевой адаптер PC
*FRAME type_name	Имя типа кадра
NODE ADDRESS address	Используется, чтобы вручную установить требуемый 16-ричный адрес PC, отличный от того, на который настроен сетевой адаптер
	Link Support
BUFFERS number [buffer_size]	Число и размер буферов для приема пакетов на PC
MAX BOARDS number	Максимальное число сетевых адаптеров, которыми может управлять модуль связи LSL.COM
MAX STACKS number	Максимальное число протоколов в стеке
MEMPOOL number[k]	Размер памяти, которую использует модуль связи LSL.COM для размещения буферов протоколов
	NetWare DOS Reguester
CACHE DUFFER SIZE = number CACHE DUFFERS = number	Размер и количество буферов в кэше на PC, который используется FIO.VLM для ввода/вывода файлов на сервер

Окончание табл. 9.2

Опция	Описание
CACHE WRITES = [ON | OFF]	ON – данные, записываемые на сетевой диск, сохраняются в кэше на PC и передаются на сервер по запросу с сервера. OFF – данные, записываемые на сетевой диск, передаются на сервер сразу
*FIRST NETWORK DRIVE = letter	Определяет первую с начала латинского алфавита букву, которая используется как логический драйв
LARGE INTERNET PACKETS=[ON| OFF]	ON – допускается использование протокола LIP
LIP START SIZE = number	Размер пакета LIP, используемый для «переговоров», когда устанавливается связь между станциями
*NAME CONTEXT = «name»	Для NetWare 4.x определяет текущий контекст при подключении пользователя к сети. Это позволяет указывать в LOGIN только конечное имя клиента
*NETWARE PROTOCOL = protocol_list	Список средств поддержки протоколов NetWare (NDS, BIND, PNW)
PB BUFFERS = number	Число буферов, используемых протоколом Packet Burst Protocol
*PREFERRED SERVER = «name»	Определяет сервер, к которому PC попытается подключиться в первую очередь
PREFERRED TREE = «name»	В NetWare 4.x определяет дерево NDS, к которому PC попытается подключиться в первую очередь
SEARCH MOSE = number	Определяет метод поиска файлов, открываемых в исполняемых файлах *.EXE и *.COM
SIGNATURE LEVEL = number	Уровень подписи NCP-пакета на PC
USE DEFAULTS=[ON| OFF]	ON – программа VLM.EXE загружает требуемые файлы *.VLM по умолчанию. OFF – файлы *.VLM не загружаются по умолчанию
VLM = path_VLM	Путь к загружаемому VLM-файлу
	Protokol IPX
IPX RETRY COUNT number	Число повторов пакета (20 по умолчанию)
IPX SOCKETS number	Максимальное число гнезд, которое может открыть станция (по умолчанию 20)

В табл. 9.3 перечислены файлы *.VLM, загружаемые программой VLM.EXE по умолчанию (USE DEFAULTS = ON).

Таблица 9.3. **Файлы *.VLM, загружаемые по умолчанию**

Файл	Описание
CONN.VLM	Обеспечивает поддержку таблиц ресурсов
IPXNCP.VLM	Обеспечивает выполнение транспортного протокола, используя IPX
TRAN.VLM	Мультиплексор транспортных протоколов стека
SECURITY.VLM	Обеспечивает дополнительную безопасность за счет возможности подписи NCP-пакетов
NDS.VLM	Обеспечивает выполнение протоколов NetWare, используя сервис NDS (NetWare 4.x)
BIND.VLM	Обеспечивает выполнение протоколов NetWare, используя сервис Bindery
NWP.VLM	Мультиплексор протоколов NetWare
FIO.VLM	Контролирует ввод/вывод данных на сетевой диск
GENERAL.VLM	Смешанные функции для файлов NETX.VLM и REDIR.VLM
REDIR.VLM	Выполняет переадресацию прерывания (21H) DOS
PRINT.VLM	Обеспечивает сервис сетевой печати
NETX.VLM	Обеспечивает совместимость с NetWare 3.11
AUTO.VLM	Обеспечивает автоматическое восстановление соединения с файловым сервером

На рис. 9.1 приведен пример файла NET.CFG

32-разрядный клиент (Client32) NetWare чаще всего устанавливается под Windows 95/98, хотя его можно инсталлировать и для DOS. Процедура установки 32-разрядного клиента под Windows 95/98 достаточно проста. Например, после запуска программы инсталляции клиента NetWare 5.x появляется несколько окон, с помощью которых необходимо выбрать язык инсталляции, платформу, куда следует установить клиентскую часть, требуемое программное обеспечение. После этого появляются окна мастера инсталляции, с помощью которых следует принять лицензионное соглашение, определить вариант установки (стандартный или заказной), выбрать сетевые протоколы (только протокол IP, только протокол IPX, IP и IPX и др.), указать предпочтительный способ подключения к сети (NDS или Bindery), определить инсталлируемые продукты (расписание выполнения необходимых работ, поддержка распределенной печати и др.).

```
Link Driver NE2000
        IRQ 5
        PORT 320
        FRAME Ethernet_802.2
NetWare DOS Requester
        FIRST NETWORK DRIVE = F
        NETWARE PROTOCOL = NDS BIND
        PREFERRED SERVER = "FS4X"
        NAME CONTEXT = "CLASS.MSTU"
```

Рис. 9.1. Пример файла NET.CFG

Пользовательская операционная среда файлового сервера

Каждый раз, когда клиент регистрируется в сети, выполняется ряд команд, настраивающих для него сетевую среду. Совокупность этих команд называется процедурой регистрации (Login Script). Эти процедуры создает администратор сети, а иногда и сам клиент.

Процедуры регистрации выполняются на PC утилитой LOGIN. После того, как эта утилита запросит имя регистрации и пароль и удостоверится, что пользователь может работать в сети, она читает процедуры регистрации и выполняет указанные в них команды.

Процедуры регистрации бывают четырех типов:

- системная,
- профильная (для NetWare 4.x/5.x),
- пользовательская,
- по умолчанию.

На рис. 9.2 представлена схема выполнения процедур регистрации в NetWare 3.x.

Системная процедура регистрации NetWare 3.12 хранится в файле SYS:PUBLIC\NET$LOG.DAT. Пользовательская процедура регистрации (файл login) хранится в подкаталоге каталога MAIL тома SYS. Этот подкаталог создается для каждого пользователя и его имя совпадает с 16-ричным идентификатором этого пользователя. Процедура регистрации по умолчанию является частью программы LOGIN и содержит команду MAP, планирующую поисковый драйв на каталог SYS:PUBLIC. Создание и модификация процедур регистрации выполняется с помощью утилиты SYSCON.

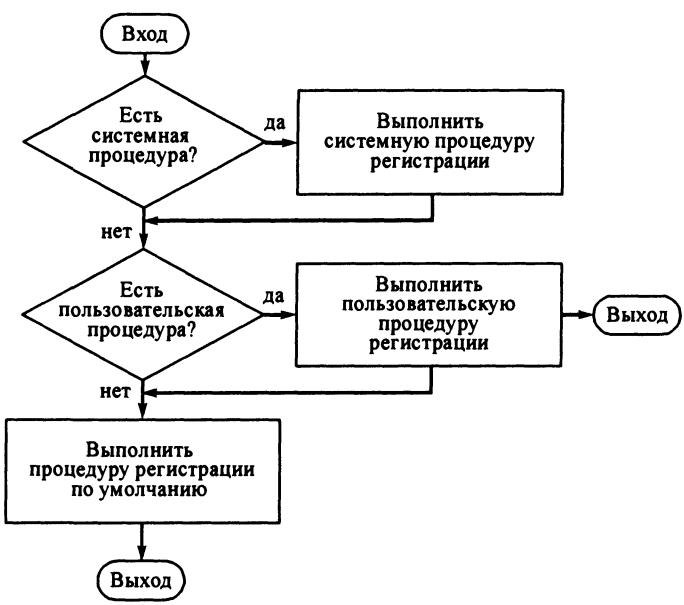


Рис. 9.2. Схема выполнения процедур регистрации в NetWare 3.12

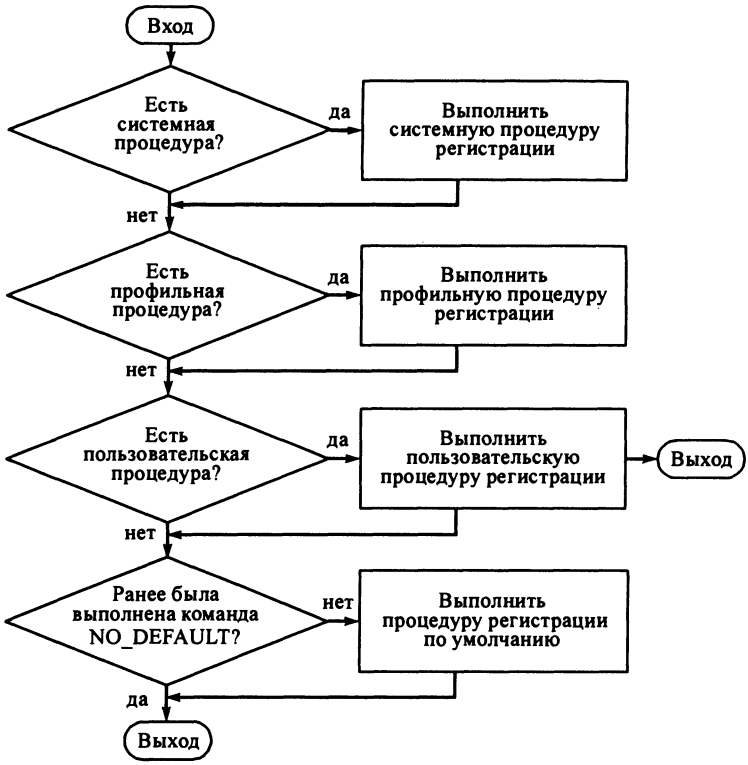


Рис. 9.3. Схема выполнения процедур регистрации в NetWare 4.x/5.x

На рис. 9.3 представлена схема выполнения процедур регистрации в NetWare 4.x/5.x.

В NetWare 4.x/5.x системная процедура регистрации хранится как свойство Login Script контейнерного объекта, непосредственно в котором описан объект пользователя (на рис. 8.54 для пользователя User это контейнер CLASS). Профильная процедура регистрации выполняется после системной, но перед пользовательской процедурой регистрации. Она описывается как свойство Login Script объекта Profile. Пользовательская процедура регистрации хранится как свойство Login Script объекта пользователя (User). В этом же свойстве можно указать на требуемый объект Profile. Процедура регистрации по умолчанию выполняется, если в системной или профильной процедуре не была выполнена команда NO_DEFAULT.

В табл. 9.4 перечислены команды, которые можно использовать в процедурах регистрации.

Таблица 9.4. **Команды процедур регистрации**

Команда	Описание
#[путь]имя_файла[параметры]	Запустить на PC резидентную программу
ATTACH[сервер[/имя[;пароль]]]	Подключиться к другому серверу 3.x, 4.x или 5.x
BREAK ON | OFF	Разрешить (ON) или запретить (OFF) пользователю прерывать процедуру регистрации по Ctrl-Break или Ctrl-C
CLS	Очистить экран PC
COMSPEC=[путь]COMMAND.COM	Указывает каталог, который будет использован DOS для повторных загрузок командного процессора (используется для удаленных PC)
CONTEXT контекст	Используется в NetWare 4.x/5.x для смены текущего контекста в дереве NDS
[F]DISPLAY [путь]имя_файла	Вывести на экран содержимое файла.
DOS BREAK [ON | OFF]	Включает (ON) или выключает (OFF) проверку нажатия клавиш Ctrl-Break и Ctrl-C при выполнении программ в DOS
[TEMP] SET переменная = «значение»	Используется для настройки переменных DOS и OS/2
DOS VERIFY [ON | OFF]	Используется для включения (ON) или отключения (OFF) режима проверки чтением после записи данных на локальные диски PC
DRIVE драйв: | *n:	Сменить текущий драйв
EXIT [«[путь]файл[параметры]»]	Прервать выполнение процедуры регистрации и запустить внешнюю программу
FIRE n	Используется для имитации звука выстрела бластера
GOTO метка ..... метка:	Продолжить выполнение процедуры регистрации с определенной метки
IF условие THEN команды [ELSE команда] END	Выполнить часть команд процедуры регистрации в зависимости от истинности условия. В условии, как правило, используются переменные процедуры регистрации (NetWare поддерживает более 30 переменных)
INCLUDE [путь]имя_файла	Включить дополнительную процедуру регистрации
LASTLOGINTIME	Используется для вывода даты и времени последней регистрации в сети

Окончание табл. 9.4

Команда	Описание
MACHINE = имя	Используется для задания имени машины (IBM_PC и т. д.)
MAP [опция] *n:=<путь|драйв:> MAP [опция] Sn:=<путь|драйв:>	Создать логическое или поисковое устройство
NO_DEFAULT	Используется в NetWare 4.x/5.x для отмены выполнения процедуры регистрации по умолчанию
NOSWAP	Предотвращает выгрузку утилиты LOGIN в расширенную память или на диск при выполнении команды #
PAUSE	По этой команде выполнение процедуры регистрации приостанавливается до нажатия клиентом любой клавиши
PCCOMPATIBLE	Если тип станции не совпадает с IBM_PC, то перед командой EXIT необходимо использовать эту команду. Применяется для удаленных PC
PROFILE имя_объекта	В NetWare 4.x/5.x определяет профильную процедуру регистрации, которая должна быть теперь выполнена
REM[ARK] [текст]	Вставить комментарий
SET TIME [ON | OFF]	Управляет синхронизацией времени сервера и PC
SHIFT [n]	Изменить нумерацию параметров, заданных в качестве аргументов утилиты LOGIN
SWAP	Разрешить выгрузку утилиты LOGIN в расширенную память или на диск при выполнении команды #
WRITE «[текст][%переменная]»	Вывести на экран сообщение

Установка и настройка сетевой ОС NetWare 3.x, 4.x/5.x

Инсталляция файлового сервера NetWare 3.x выполняется с дискет, а NetWare 4.x/5.x – с CD-ROM (как правило). Установка NetWare 4.x/5.x проще, чем инсталляция NetWare 3.x. Некоторые специалисты считают, что одним из недостатков NetWare является более сложная процедура установки ОС, чем в Windows NT. Рассмотрим основные шаги инсталляции NetWare 3.x, NetWare 4.x и NetWare 5.x.

Ниже описана процедура установки NetWare 3.x на файловом сервере.

1. Создать раздел DOS (5 Мб).

2. Скопировать дискеты NetWare 3.x Operating System – 1, 2, 3 в директорию DOS.

3. Сделать эту директорию текущей и запустить головную программу ОС SERVER.EXE.

4. Задать имя сервера и внутренний номер сети.

5. С помощью команды LOAD загрузить драйвер жесткого диска (*.DSK).

6. По команде LOAD запустить NLM-модуль INSTALL. NLM.

7. Создать на жестком диске раздел NetWare (пункт Disk Options).

8. Создать и смонтировать тома раздела NetWare (пункт Volume Options).

9. Скопировать файлы на том SYS (пункт System Options).

10. Переключиться с помощью клавиши ALT-ESC на экран консоли файлового сервера.

11. Загрузить драйвер сетевого адаптера. Например:

LOAD имя_драйвера INT=номер PORT=адрес

Присоединить драйвер сетевого адаптера к протоколу IPX. Например:

BIND IPX TO имя_драйвера

12. Переключиться с помощью клавиши ALT-ESC обратно на экран утилиты INSTALL.NLM.

13. Создать два файла автоматической загрузки AUTOEXEC.NCF и STARTUP.NCF (пункт System options).

14. Выйти из утилиты INSTALL.NLM и с помощью команды DOWN завершить работу с сервером.

Рассмотрим процедуру установки NetWare 4.1. В отличие от предыдущей процедуры здесь не требуется, чтобы администратор помнил последовательность выбора пунктов меню, вводить команды с консоли файлового сервера и переключаться с экрана на экран. Можно выполнить простую (simple) и заказную (custom) инсталляцию NetWare 4.1. Ниже приведены шаги заказной инсталляции (Custom Installation).

1. Создать загрузочный раздел DOS и установить программное обеспечение для работы с CD-ROM в DOS.

2. Вставить диск с ОС NetWare 4.1 в устройство CD-ROM и запустить файл INSTALL.BAT.

3. Из меню выбрать язык, на который настраивается файловый сервер.

4. Выбрать пункт меню NetWare Server Installation, а затем пункт NetWare 4.1.

5. Выбрать пункт меню Custom Installation of NetWare 4.1.

6. Задать имя сервера и внутренний номер сети.

7. Подтвердить копирование файлов загрузки в раздел DOS.

8. Специфицировать код страны, кодовую страницу, тип клавиатуры и нажать клавишу F10.

9. Выбрать формат имени файла (рекомендуется как в DOS).

10. Выбрать YES, если необходимо вручную задать команды файла STARTUP.NCF.

11. Выбрать YES, если необходимо добавить строку SERVER.EXE в AUTOEXEC.BAT.

После этого программа инсталляции автоматически загружает SERVER.EXE, INSTALL.NLM и предлагает выполнить требуемые пункты меню.

12. Выбрать из списка драйвер диска.

13. Выбрать драйвер сетевого адаптера и указать номер прерывания и адрес порта ввода/вывода.

14. Создать на жестком диске раздел NetWare.

15. Создать и смонтировать тома раздела NetWare.

16. Вставить дискету с лицензией (License diskette) в устройство A:.

17. С помощью клавиши Enter инициировать задание параметров для копирования файлов на том SYS.

18. Выбрать группы копируемых файлов (OS/2, Workstation Utilities и т. д.) и нажать клавишу F10, чтобы стартовать копирование файлов на том SYS.

19. Инсталлировать дерево NDS:

- ввести имя дерева NDS,
- определить параметры синхронизации времени (тип временного сервера, временную зону, параметры перехода на зимнее и летнее время),
- определить контекст сервера (имя объекта типа O, где будет создан объект администратора, и имя объекта типа OU, где будут созданы объекты сервера и его томов),
- указать имя администратора (по умолчанию Admin) и его пароль.

20. При необходимости модифицировать файлы STARTUP.NCF и AUTOEXEC.NCF.

21. Из предлагаемого списка Choose an Item of Product Listed Above выбрать необходимые продукты и проинсталлировать их (например, электронную документацию Dynatext, шлюз электронной почты MHS и т. д.).

22. Завершить работу файлового сервера.

Ниже кратко описана процедура установки NetWare 5.

1. Создать загрузочный раздел DOS и установить программное обеспечение для работы с CD-ROM в DOS. Система NetWare 5 поставляется на лазерном диске, с которого можно загрузить компьютер. При этом автоматически запускается программа инсталляции, позволяющая при необходимости выполнить указанные выше действия. В дальнейшем предполагается, что загрузка DOS выполнена с жесткого диска.

2. Перейти на устройство CD-ROM и в командной строке ввести Install. Сразу после запуска программа инсталляции загружает сервер, и весь дальнейший процесс установки происходит под управлением ОС NetWare 5.

3. Выбрать язык, на котором будет выполнена установка (обычно English), согласиться с лицензионным соглашением.

4. Выбрать тип инсталляции (New Server или Upgrade), каталог DOS для загрузочных файлов NetWare (обычно C:\NWSERVER).

5. Выбрать страну, кодовую таблицу, раскладку клавиатуры (лучше все для USA).

6. Выбрать тип мыши и видеоадаптера (обычно Super VGA).

7. Уточнить модули поддержки платформы и контроллер дисков (программа инсталляции обычно распознает эти параметры автоматически).

8. Уточнить параметры драйверов дисков и сетевых адаптеров (программа инсталляции обычно определяет эти параметры автоматически).

9. Удалить существующий раздел NetWare (для типа инсталляции New Server, см. п. 4), создать новый раздел NetWare и том SYS.

Далее инсталляция продолжается в графическом режиме.

10. Определить имя сервера.

11. Создать дополнительные тома.

12. Выбрать протокол, по которому должен работать сервер (IP, IPX, IP и IPX).

13. Указать часовой пояс.

14. Инсталлировать новое дерево NDS или включить сервер в уже существующее дерево.

15. Установить лицензию, поставляемую на дискете.

16. При необходимости установить на сервере дополнительные продукты.

9.2. Администрирование информационной среды

Категории пользователей сетевой ОС NetWare

Существуют четыре уровня ответственности, которые могут быть присвоены пользователям NetWare:

- рядовые пользователи сети,
- операторы (операторы консоли файлового сервера, операторы очереди печати, операторы сервера печати),
- менеджеры (руководители групп),
- администраторы сети.

Рядовые пользователи – пользователи, работающие в сети. Они могут запускать прикладные программы и работать с файлами в соответствии с теми правами, которыми они наделены.

Операторы – рядовые пользователи, которые наделены дополнительными привилегиями. Например, оператор консоли файлового сервера – это пользователь, наделенный правом использования утилиты NetWare 3.12 FCONSOLE, которая позволяет контролировать работу сервера (просматривать время загрузки сервера, анализировать, могут ли подключаться новые пользователи и включена ли TTS) и выгружать сервер.

Менеджеры – пользователи, имеющие право создания новых объектов пользователей и управления ими. Менеджеры рабочих групп (Workgroup Managers) могут выполнять обе эти функции, а менеджеры контроля пользователей (User/Group Account Managers) – только управлять пользователями. Менеджеры работают как администраторы, но только в пределах группы.

В NetWare 3.x менеджеры назначаются администратором с помощью утилиты SYSCON. В NetWare 4.x/5.x для этого используется объект Organizational Role. В качестве свойства объекта Organizational Role администратором назначается список «оккупантов» этого объекта. При этом объект Organizational Role автоматически включается в список Security Equal To всех «оккупантов». Как правило, объект Organizational Role назначает опекун управляемого контейнерного объекта. Если объекту Organizational Role назначить супервизорное право [S], то «оккупант» может лишить администратора права управлять этим контейнерным объектом (см. §. 8.6, табл. 8.20).

Администраторы сети ответственны за нормальное функционирование всей сети. В их функции входит обслуживание сети, а также осуществление изменений в сети и ее реконфигурирование:

- создание счетов (т. е. объектов пользователей),
- создание и реконфигурирование серверов печати и очередей,
- архивация и восстановление данных,
- назначение опекунов файловой системы и NDS,
- создание групп и их менеджеров,
- инсталляция дополнительного сервера и т. д.

Ведение системной базы данных сетевых ресурсов

В NetWare 3.x СБДСР называется Bindery, а в NetWare 4.x/5.x – NDS (NetWare Directory Service). СБДСР Bindery включает в себя объекты, их свойства и значения свойств. Объектом является любая физическая или логическая единица, зарегистрированная в сети, т. е. все, что имеет имя. К объектам сети относятся пользователи, группы пользователей, файловые серверы, серверы печати, очереди вывода на печать. Каждому объекту при создании присваивается идентификационный номер (ID). В NetWare 3.x база данных сетевых ресурсов состоит из трех файлов, которые располагаются в каталоге SYS:SYSTEM и имеют атрибут Hidden:

- NET$OBJ.SYS – список объектов,
- NET$PROP.SYS – списки свойств, которыми обладают объекты,
- NET$VAL.SYS – наборы значений всех свойств объектов.

Доступ к объектам БД Bindery осуществляется с помощью утилит-меню SYSCON и PCONSOLE. База данных Bindery может быть сохранена и восстановлена с помощью средств копирования и архивирования файловой системы (NCOPY, SMS и т. д.). Для восстановления разрушенных файлов Bindery в NetWare 3.x используют утилиту BINDFIX.

База данных NDS также, как и БД Bindery, включает объекты, их свойства и значения свойств, но состав объектов NDS и их свойств значительно шире. Организация NDS будет рассмотрена ниже. В NetWare 4.x/5.x база данных сетевых ресурсов хранится в скрытом каталоге SYS:_NETWARE, в который входят следующие файлы:

- ENTRY.NDS – содержит описание объектов NDS и их свойств,

• VALUE.NDS и BLOCK.NDS – содержат значения свойств объектов,
• *.000 – эти файлы содержат потоковые значения свойств (Login Script, Print Job Configuration и т. д.),
• MSL.000 – содержит лицензию NetWare 4.x/5.x.

Основным средством администрирования базы данных NDS является Windows-программа NetWare Administrator (NWADMIN.EXE, NWADMN95.EXE – для NetWare 4.x, NWADMN32.EXE – для NetWare 5.x). Доступ к БД NDS можно осуществить и с помощью текстовой утилиты-меню NETADMIN, но она имеет небогатые «выразительные» возможности. База данных NDS может быть сохранена и восстановлена с помощью средства архивирования SMS (Storage Management Services). В NetWare 4.x/5.x предусмотрено несколько утилит файлового сервера для работы с базой данных NDS:

• DSMERGE.NLM – объединить два разных дерева NDS в одно дерево *NDS*; эта утилита важна, если к одной сети подключаются два сервера, которые раньше работали в разных сетях,
• DSREPAIR.NLM – «отремонтировать» базу данных NDS.

Сопровождение сетевой файловой системы

В NetWare 3.12 и 4.x/5.x архивация и восстановление информации обеспечивается при помощи специального средства SMS (Storage Management Services). SMS позволяет администратору архивировать следующую информацию:

• файловую систему сервера NetWare 3.x и 4.x/5.x;
• базу данных NDS (для NetWare 4.x/5.x);
• файловую систему DOS-рабочей станции;

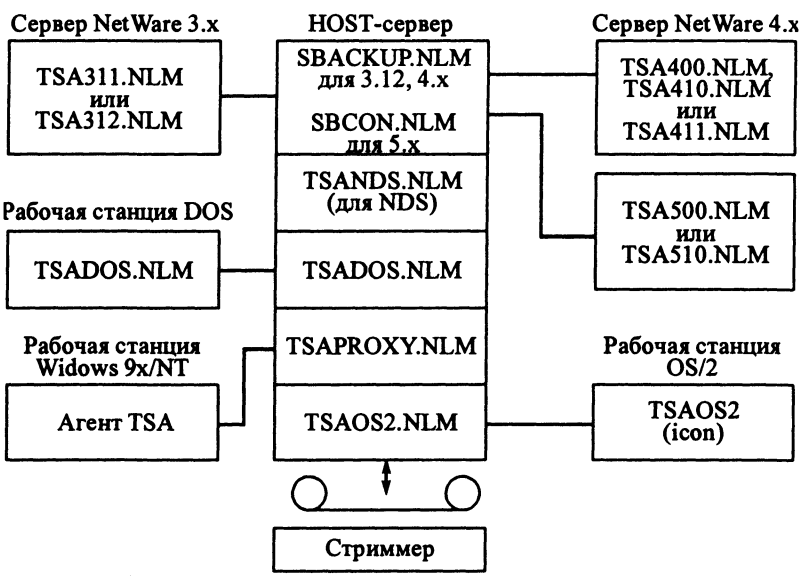


Рис. 9.4. Модули, загружаемые на host-сервере и target-устройствах

- файловую систему Windows 9x-рабочей станции;
- файловую систему OS/2-рабочей станции;
- базы данных Btrieve.

В процессе архивации участвуют host-сервер (сервер NetWare, на котором выполняется программа архивации SBACKUP.NLM) и target-устройство (сервер NetWare или рабочая станция, которые содержат данные для архивации). На рис. 9.4 представлены модули, которые должны быть загружены на host-сервере и target-устройствах.

Необходимо отметить, что при начальной инсталляции PC (INSTALL.EXE) следует указать на возможность архивирования файловой системы этой PC.

SMS поддерживает четыре способа архивации (табл. 9.5). Эти способы используют в различных комбинациях, определяющих стратегии архивации (табл. 9.6).

Таблица 9.5. **Способы архивации**

Способ архивации	Архивируемые данные	Состояние бита модификации (Modify Bit)
Full	Все данные независимо от того, архивировались ли они ранее	Стирается
Incremental	Файлы, которые были созданы или модифицированы после последней Full- или Incremental-архивации	Стирается
Differential	Все файлы, которые были созданы или модифицированы после последней Full-архивации	Не стирается
Custom	Только те данные, которые определил администратор	Определяется администратором

Таблица 9.6. **Стратегии архивации**

Стратегия архивации	Время, необходимое для архивации	Время, необходимое для восстановления
Full-архивация	Максимальное	Минимальное
Full-архивация с последующей Incremental-архивацией.	Минимальное	Максимальное
Full-архивация с последующей Differential-архивацией	Промежуточное между двумя предыдущими стратегиями	Промежуточное между двумя предыдущими стратегиями
Custom-архивация	Определяется объемом сохраняемых данных	Определяется объемом восстанавливаемых данных

При выборе стратегии архивации следует учитывать время, необходимое для архивации и восстановления данных.

Ниже перечислены основные шаги, которые должен выполнить администратор сети при выполнении архивации данных (NetWare 3.12 и 4.x).

1. Запустить на host-сервере драйвер устройства архивации (стримера).
2. Загрузить модули на требуемых target-устройствах (см. рис. 9.4).
3. Загрузить SBACKUP.NLM на host-сервере. При этом автоматически загружается модуль SMDR.NLM.
4. Выбрать в окне SBACKUP.NLM target-устройство, данные которого будут архивироваться (сервер, БД NDS, файловую систему PC DOS и OS/2, БД Btrieve).
5. Если необходимо, то следует подключиться из SBACKUP.NLM к требуемому target-устройству.
6. Выбрать местоположение для файла, содержащего журнал сеанса архивации (по умолчанию SYS:SYSTEM\TSA\LOG).
7. Выбрать тип архивации (см. табл. 9.5).
8. Присвоить дескриптор для данного сеанса архивации.
9. При необходимости выбрать опцию, добавляющую сеанс архивации к предыдущему.
10. Для Custom-архивации определить данные для архивации (часть дерева NDS, часть файловой системы).
11. Выбрать время архивации: теперь или позже.
12. Вставить магнитный носитель архива (стример).
13. Ввести метку для нового набора данных.
14. Выполнить архивацию.

Ниже перечислены основные шаги, которые должен выполнить администратор при восстановлении данных из архива (NetWare 3.12 и 4.x).

1. Запустить на host-сервере драйвер устройства архивации (стримера).
2. Загрузить модули на требуемых target-устройствах (см. рис. 9.4).
3. Загрузить SBACKUP.NLM на host-сервере.
4. Выбрать в окне SBACKUP.NLM target-устройство, данные которого нужно восстановить.
5. Если необходимо, то следует подключиться из SBACKUP.NLM к требуемому target-устройству как пользователь, имеющий требуемые права для выполнения восстановления.
6. Выбрать режим Restore a Session (восстановление сеанса).
7. Определить директорию, содержащую журнал сеанса архивации.
8. Выбрать необходимый сеанс.
9. Вставить магнитный носитель, содержащий восстанавливаемые данные.
10. Выбрать тип восстановления:
- восстановление полного сеанса архивации,
- Custom-восстановление.

11. Для Custom-восстановления определить восстанавливаемые данные.
12. Выбрать время восстановления: теперь или позже.

В NetWare 5.x резервное копирование и восстановление можно выполнять как с экрана сервера (утилита SBCON), так и с PC (утилита NWBACK32.EXE). Утилита NWBACK32.EXE конкурирует с известными средствами архивирования фирмы Cheyenne ARCserve.

Обзор средств администратора

В табл. 9.7 перечислены основные утилиты администратора NetWare 3.12, а в табл. 9.8 – утилиты администратора NetWare 4.x/5.x. Эти утилиты следует запускать на PC.

Таблица 9.7. **Утилиты администратора NetWare 3.12**

Утилита	Описание
	Графические утилиты, выполняемые под Windows 3.1
NWUSER	Используется для динамического планирования логических драйвов, портов печати, а также для просмотра ресурсов сети и посылки сообщений
DTEXTRW	Используется для просмотра электронной документации
	Утилиты-меню, выполняемые под DOS
COLORPAL	Позволяет изменить цвета уже существующих палитр элементов окон утилит-меню NetWare. 312
DSPACE	Позволяет ограничить дисковое пространство, доступное клиентам, и дисковое пространство в каталоге
FCONSOLE	Позволяет операторам консоли файлового сервера посылать сообщения всем клиентам, просматривать информацию о соединениях клиентов, останавливать сервер (DOWN)
FILER	Позволяет выполнить защиту каталога или файла и операции с каталогом или файлом (удаление, копирование)
MAKEUSER	Позволяет создать или удалить большое число клиентов с одинаковыми правами
RCONSOLE (в SYSTEM)	Позволяет организовать удаленную консоль
SALVAGE	Позволяет восстановить ранее удаленные (по delete) файлы
SESSION	Позволяет создать логические и поисковые драйвы
SYSCON	Используется для создания объектов пользователей, групп и присвоения им опекунских прав. Позволяет создавать и изменять процедуры регистрации, просматривать бюджеты, ограничивать регистрацию и т. д.
USERDEF	Используется администратором сети или менеджером рабочих групп для создания нескольких пользователей. Эта утилита формирует файл *.USR для автоматического создания клиентов при помощи утилиты MAKEUSER
VOLINFO	Позволяет посмотреть информацию о заполнении томов файлового сервера

Продолжение табл. 9.7

Утилита	Описание
	Утилиты командной строки, выполняемые под DOS
ALLOW	Позволяет устанавливать и модифицировать фильтры наследуемых прав (IRF) для каталогов и файлов
ATTACH	Используется для подключения к другим файловым серверам
BINDFIX	Используется, чтобы «отремонтировать» файлы базы данных Bindery
BINDREST	Используется для реставрации предыдущих файлов базы данных Bindery, которые сохраняются утилитой BINDFIX (файлы *.OLD)
CASTOFF	Запретить прием сообщений на PC
CASTON	Разрешить прием сообщений
CHKDIR	Позволяет отобразить информацию о заполнении тома и директории
CHKVOL	Позволяет отобразить информацию о характеристиках тома и о пространстве, занимаемом удаленными файлами
DOSGEN	Создает файл удаленной загрузки для бездисковых станций
FLAG	Используется для просмотра и изменения атрибутов файлов
FLAGDIR	Используется для просмотра и изменения атрибутов каталогов
GRANT	Используется для назначения опекунских прав клиентам и группам по отношению к каталогу или файлу
LISTDIR	Используется для просмотра эффективных прав клиента и фильтра наследуемых прав во всех подкаталогах текущего каталога
LOGIN	Используется для регистрации клиента на файловом сервере
LOGOUT	Используется для отключения клиента от одного или всех файловых серверов
MAP	Используется для просмотра, назначения или отмены логического или поискового драйва
MENUMAKE	Позволяет оттранслировать описание пользовательского меню
NCOPY	Используется для копирования файлов из одного каталога в другой
NDIR	Применяется для просмотра разнообразной информации о файлах и каталогах
NMENU	Позволяет запустить пользовательское меню
NVER	Используется для просмотра версии ПО, загруженного на файловом сервере и PC
PAUDIT (в SYSTEM)	Применяется для просмотра записей системного бюджета
PURGE	Используется для очистки каталога от ранее удаленных (по delete) файлов
REMOVE	Используется для исключения пользователя или группы из списка опекунов какого-либо каталога или файла
RENDIR	Применяется для переименования каталога

Окончание табл. 9.7

Утилита	Описание
RENDIR	Применяется для переименования каталога
REMOKE	Используется для отмены у пользователя или группы всех или некоторых опекунских прав на каталог или файл
RIGHTS	Применяется для просмотра действительных (эффективных) прав клиента по отношению к каталогу или файлу
SECURITY (в SYSTEM)	Оповещает о потенциальных нарушителях в сети
SEND	Послать сообщение с PC одному или всем активным клиентам сети
SETPASS	Позволяет установить или изменить пароль на файловом сервере
SETTTS	Устанавливает пороги для транзакционных файлов
SLIST	Показывает список активных файловых серверов
SMODE	Используется для назначения исполняемому файлу режима поиска файлов, открываемых в этом исполняемом файле
SYSTIME	Позволяет посмотреть дату и время, установленные на файловом сервере
TLIST	Используется для просмотра списка опекунов файла или каталога
USERLIST	Используется для просмотра списка активных пользователей
VERSION	Используется для просмотра версии утилиты или NLM-модуля
WHOAMI	Используется для просмотра информации о клиенте, работающем на текущей станции
WSUPDATE (в SYSTEM)	Применяется для обновления файлов рабочей станции с файлового сервера
PRINTDEF, PCONSOLE, PRINTCON, CAPTURE	Используются для настройки сетевой печати
NPRINT	Используется для передачи задания в очередь на печать

Таблица 9.8. **Утилиты администратора NetWare 4.x/5.x**

Утилита	Описание
Графические утилиты, выполняемые под Windows 95/98/NT	
NWADMIN, NWADMN95 (4.x), NWADMN32 (5.x)	Основная утилита администратора в NetWare 4.x/5.x для работы с деревом NDS
NWUSER (4.x)	Используется для динамического планирования логических драйвов, портов печати, а также для просмотра ресурсов сети и посылки сообщений

Продолжение табл. 9.8

Утилита	Описание
Утилиты-меню, выполняемые под DOS	
AUDITCON	Позволяет организовать аудиторскую проверку данных
COLORPAL (4.x)	Позволяет изменить цвета уже существующих палитр элементов окон утилит-меню
FILER (4.x)	Предназначена для управления доступом к файлам и каталогам, а также для восстановления ранее удаленных файлов
NETADMIN (4.x)	Текстовая утилита для работы с деревом NDS
NETUSER (4.x)	Текстовая утилита для динамического планирования логических и поисковых драйвов, портов печати, а также для просмотра ресурсов сети и посылки сообщений
PARTMGR (4.x), NDSMGR32 (5.x)	Выполнять операции с разделами и репликами дерева NDS
RCONSOLE	Позволяет организовать удаленную консоль
Утилиты командной строки, выполняемые под DOS	
ATOTAL	Позволяет получить информацию о том, какие услуги сервера можно сделать платными и предварительно оценить размеры тарифов
CX	Позволяет просмотреть и изменить текущий контекст в дереве *NDS*
DOSGEN (4.x)	Создает файл удаленной загрузки для бездисковых станций
FLAG (4.x)	Используется для просмотра и изменения атрибутов каталогов и файлов
LOGIN	Используется для регистрации клиента в дереве NDS
LOGOUT	Используется для отключения клиента от сети
MAP	Применяется для просмотра, назначения или отмены логического или поискового драйва
MENUMAKE (4.x)	Позволяет оттранслировать описание пользовательского меню
NCOPY	Используется для копирования файлов из одного каталога в другой
NDIR	Применяется для просмотра разнообразной информации о файлах и каталогах
NLIST	Используется для просмотра списка объектов, свойств и их значений в дереве NDS
NMENU (4.x)	Позволяет запустить пользовательское меню
NVER	Используется для просмотра версии ПО, загруженного на файловом сервере и PC
NWXTRACT (4.x)	Копирует файлы с CD-ROM на сетевой или локальный диск

Окончание табл. 9.8

Утилита	Описание
PURGE	Применяется для очистки каталога от ранее удаленных (по delete) файлов
RENDIR	Применяется для переименования каталога
RIGHTS	Посмотреть или модифицировать права пользователя или группы по отношению к каталогу или файлу
SEND (4.x)	Используется для посылки сообщения PC одному или всем активным клиентам сети
SETPASS	Позволяет установить или изменить пароль клиента
SETTTS (4.x)	Устанавливает пороги для транзакционных файлов
SYSTIME (4.x)	Выполняет синхронизацию времени PC и сервера
UIMPORT (4.x)	Добавляет объекты в дерево NDS из ASCII-файла
WHOAMI (4.x)	Используется для просмотра информации о клиенте, работающем на текущей станции
WSUPDATE (4.x)	Применяется для обновления файлов PC с файлового сервера
WSUPGRD (4.x)	Обновляет драйвер сетевого адаптера на PC
Для 4.x: PRINTDEF, PCONSOLE, PRINTCON, CAPTURE	Используются для настройки сетевой печати
PSC (4.x)	Обеспечивает контроль за сервером печати и принтерами
NPRINT	Используется для передачи задания в очередь на печать

9.3. Управление сетевыми ресурсами

Глобальный сетевой каталог NDS

В ОС NetWare 4.x/5.x системная база данных сетевых ресурсов называется NDS (NetWare Directory Services) и она существенно отличается от базы данных Bindery в NetWare 3.x. Наиболее важными отличиями NDS от Bindery являются следующие:

- в базу данных можно включить много новых объектов,
- объекты базы данных иерархически связаны между собой, такую структуру называют деревом NDS,
- NDS – это база данных всей сети, а не отдельного сервера,
- в NetWare 4.x/5.x предлагается графическая утилита для работы с базой данных NDS (Windows-программа NetWare Administrator).

Таким образом, в NetWare 4.x/5.x логические и физические ресурсы сети должны быть описаны администратором в виде дерева NDS. На верхних уров-

нях дерева (ближе к корню) располагаются описания логических ресурсов, которые принято называть *контейнерными объектами*; на самых нижних (листьевых) уровнях – описания физических ресурсов, которые также называют *оконечными объектами*. В контейнерных объектах можно создавать другие объекты, в оконечных – нельзя.

В табл. 9.9 приведен список контейнерных объектов.

Таблица 9.9. **Контейнерные объекты**

Имя объекта	Обозначение	Примечания
Root		Корень дерева
Country	C	Страна
Organization	O	Организация
Organizational Unit	OU	Организационная единица (отдел, цех, комната и т. д.)

Общие правила формирования дерева NDS представлены на рис. 9.5. В объекте Root можно создавать объекты типа Country и Organization. Объект Country может включать только объекты типа Organization, в которых могут размещаться объекты Organizational Unit и оконечные объекты (Leaf). В объекте Organizational Unit также можно создавать объекты типа Organizational Unit и Leaf.

Интерпретация контейнерных объектов может быть самой различной. Это могут быть: регионы, предприятия, отделения, цеха, отделы, участки, комнаты и т. д.

В табл. 9.10 перечислены некоторые наиболее важные оконечные объекты (Leaf), которые можно включать в контейнерные объекты.

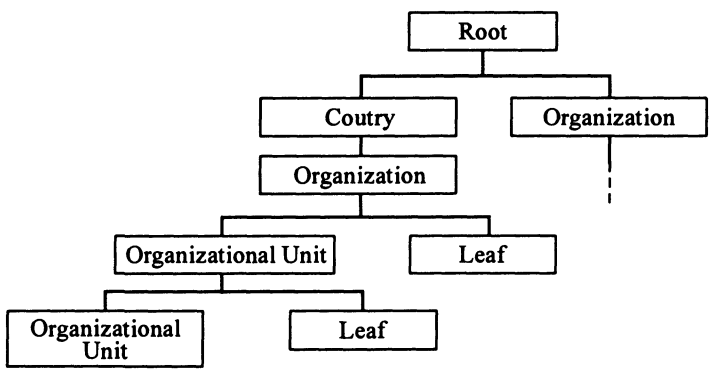


Рис. 9.5. Правила формирования дерева NDS

Таблица 9.10. **Оконечные объекты дерева NDS**

Имя объекта	Примечания
AFP Server	Сервер, базирующийся на протоколе Appletalk (для ОС Macintosh)
Alias	Ссылка на действительно существующий объект
Computer	Рабочая станция или маршрутизатор (используется для хранения адреса станции, имени пользователя и т. д.)
Directory Map	Ссылается на какую-либо директорию файловой системы (может использоваться в Login Script при кодировании команды MAP)
Group	Именует список (группу) объектов User (права, назначенные группе, автоматически передаются объектам User из списка)
Organizational Role	Определяет менеджера для контейнера
Print Server	Определяет сервер печати
Printer	Описывает сетевой принтер
Printer Queue	Определяет очередь заданий на печать
Profile	Позволяет создать Login Script, который может выполняться после системного сценария подключения, но перед пользовательским
Netware Server	Описывает файловый сервер NetWare. Он автоматически создается при инсталляции файлового сервера и на этот объект можно ссылаться в свойствах объектов дерева NDS и в утилитах NetWare 4.x/5.x
Template	Определяет шаблон, который можно использовать при включении в дерево NDS объектов типа User
User	Описывает пользователя, который может регистрироваться в сети
Volume	Описывает физический том файлового сервера (на этот объект можно ссылаться в свойствах других объектов и в утилитах NetWare 4.x/5.x)
Directories	Описывает директорию (для объектов Volume и Directories). Этот объект не хранится в дереве NDS, а читается из структуры файловой системы программой, обслуживающей дерево NDS

Листьевые объекты имеют одно системное обозначение – CN. Каждый из них имеет свой набор свойств.

В результате инсталляции файлового сервера автоматически создается дерево (рис. 9.6). При установке NetWare 5.x в NDS автоматически добавляются и некоторые другие объекты (на рис. 9.6 не показаны).

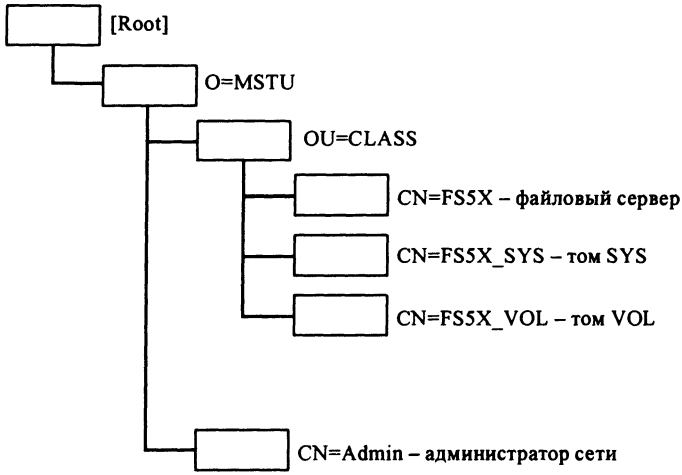


Рис. 9.6. Пример дерева, создаваемого при инсталляции файлового сервера

При инсталляции дерева NDS администратор должен указать контекст для размещения новых объектов Server и Volume: имена объектов Organization (O) и Organizational Unit (OU). С помощью утилиты NetWare Administrator администратор может расширить дерево NDS по своему усмотрению.

Наличие службы NDS является весомым преимуществом ОС NetWare 4.x/5.x. Используемая в ОС Windows 2000 служба каталогов Active Directory по своим возможностям уступает NDS.

Манипулирование сетевыми ресурсами

После запуска в Windows утилиты NetWare Administrator на экране появляется окно с изображением дерева NDS (рис. 9.7).

Слева от каждого имени объекта высвечивается знак, указывающий на класс, к которому принадлежит этот объект. Классы объектов (см. табл. 9.10) показаны на рис. 9.8.

Над объектами в дереве NDS можно выполнять следующие операции:

- осуществлять поиск объектов в дереве NDS,
- создавать новые объекты,
- модифицировать свойства объектов,
- перемещать объекты в дереве NDS,
- удалять объекты.

Рассмотрим эти операции подробнее:

Поиск объектов в дереве NDS. Чтобы посмотреть содержимое контейнерного объекта, необходимо дважды щелкнуть мышью на имени этого контейнера в окне Browse (см. рис. 9.7). А чтобы скрыть содержимое следует повторно дважды щелкнуть мышью на имени контейнерного объекта. Поиск объекта с заданными свойствами можно выполнить, выбрав пункт меню Object/Search утилиты NetWare Administrator.

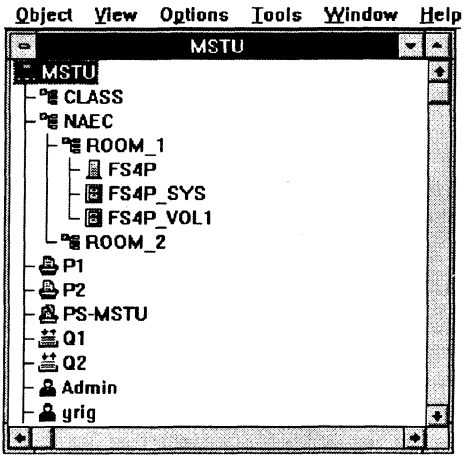


Рис. 9.7. Фрагмент дерева NDS

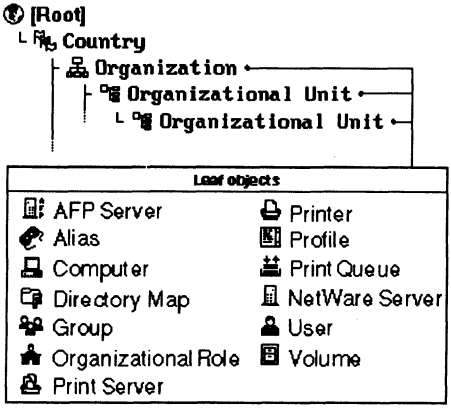


Рис. 9.8. Классы объектов

Создание объектов. Для создания нового объекта необходимо в дереве NDS (см. рис. 9.7) указать мышью на контейнерный объект, куда следует включить новый объект, и выбрать пункт меню Object/Create утилиты NetWare Administrator. Из появившегося меню нужно выбрать класс включаемого объекта (см. рис. 9.8) и заполнить поля окна, имеющего примерно следующую структуру (рис. 9.9).

Если указать второй признак, то после создания объекта система предложит создать другой объект того же типа. Первый признак является альтернативой второму. Если он указан, то после нажатия кнопки OK на экране появится окно, имеющее следующую структуру (рис. 9.10). Справа располагается меню свойств, слева – поля со значениями свойства, выбранного из меню. Каждый объект имеет свой набор свойств. В частности, многие объекты имеют свойство Rights to Files and Directories, с помощью которого объект можно

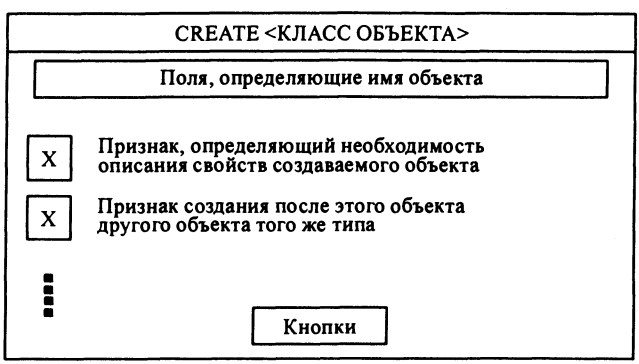


Рис. 9.9. Структура окна с именем объекта

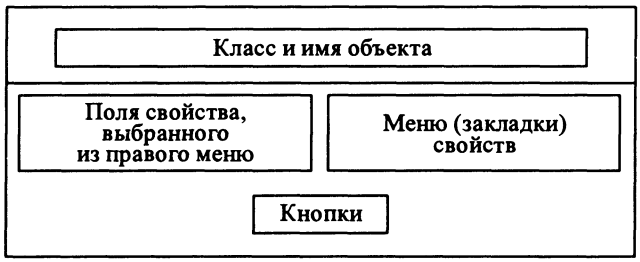


Рис. 9.10. Окно экрана для определения значений свойств

сделать опекуном нескольких директорий или файлов. Объект типа Directories обладает свойствами Trustees of this Directory и Directory Attributes, которые позволяют определить фильтр наследуемых прав (IRF), описать несколько опекунов к этой директории и назначить атрибуты. Дважды щелкнув мышью на имени директории в дереве NDS (см. рис. 9.7), можно посмотреть поддиректории и файлы, хранящиеся в ней. Они имеют примерно тот же самый набор свойств.

После определения значений свойств и нажатия кнопки OK на экране вновь появляется окно Browse (см. рис. 9.7). Чтобы переименовать объект, следует подсветить его в дереве NDS и выбрать пункт меню Object/Rename.

Модификация свойств объекта. Чтобы модифицировать свойства объекта, необходимо подсветить его в дереве NDS, нажать правую клавишу мыши и в появившемся меню выбрать пункт Details (табл. 9.11). На экране появится окно для определения значений свойств объекта (рис. 9.10).

Таблица 9.11. **Описание пунктов контекстного меню, которое появляется после нажатия правой клавиши мыши**

Пункт меню	Описание
Details	Отобразить свойства объекта
Rights to other Objects	Показать список объектов, для которых текущий объект является опекуном
Trustees of this Object	Отобразить список опекунов выбранного объекта. С помощью этого пункта можно определить фильтр наследуемых прав (IRF) и описать несколько опекунов к этому объекту (см. § 8.6)
Browse	Показать на экране новое окно Browse
Create	Создать новый объект в текущем контейнерном объекте
Delete	Удалить текущий объект

Перечисленные в табл. 9.11 пункты меню (кроме Browse) можно также найти в меню Object.

Перемещение объектов в дереве NDS. Для перемещения *оконечных* объектов в другой контейнерный объект необходимо указать мышью на перемещаемый объект (чтобы указать несколько объектов, следует удерживать клавишу Ctrl или Shift), выбрать пункт Object/Move (или нажать клавишу F7) и выбрать контейнерный объект, куда следует переместить выбранные оконечные объекты.

Для перемещения *контейнерного* объекта вместе со своими объектами сначала с помощью утилиты NDS Manager Utility следует создать раздел для этого контейнера и только потом выполнить его перемещение в другой контейнерный объект.

Удаление объектов в дереве NDS. Для удаления объектов необходимо указать мышью на удаляемый объект (чтобы указать несколько объектов, следует удерживать клавишу Ctrl или Shift), выбрать пункт меню Object/Delete (или нажать клавишу Delete) и подтвердить необходимость удаления объектов. Контейнерный объект можно удалять только в том случае, если он пуст.

Процедура однократного подключения пользователя к сети

Подключившись по LOGIN к сети, пользователь получает доступ ко всем файловым серверам, их томам и другим объектам, по отношению к которым он имеет соответствующие права. При выполнении команды LOGIN пользователь прежде всего должен указать имя регистрации. По этому имени NetWare 4.x/5.x определяет местоположение требуемого объекта User в дереве NDS. В табл. 9.12 определяются правила кодирования имени регистрации в команде LOGIN.

Таблица 9.12. **Правила кодирования имени регистрации в команде LOGIN**

Вариант кодирования	Описание
Имя1[.Имя2...]	Наличие точки в начале введенной строки означает, что это «Имя1[.Имя2...]» определяет полное имя объекта User в дереве NDS (т. е. полный путь от объекта User с именем «Имя1» до корня дерева)
Имя1[.Имя2...]	Отсутствие точки в начале и конце введенной строки означает, что для определения местоположения объекта User в дереве NDS процедура подключения присоединяет слева эту строку к имени контекста, указанному администратором при инсталляции PC. Если имя контекста PC имеет вид «Name1[.Name2...]», то полный путь поиска определяется строкой «Имя1[.Имя2...].Name1 [.Name2...]»
Имя1[.Имя2...].	Наличие точки в конце введенной строки означает, что для определения местоположения объекта User эта строка присоединяется слева к имени контекста PC, расположенному на более высоком уровне иерархии, т. е. если имя контекста рабочей станции имеет вид «Name1[.Name2...]», то полный путь поиска определяется строкой «Имя1[.Имя2...][.Name2...]»

Имена в команде LOGIN и в строке Name Context могут быть составными, т. е. могут состоять из идентификаторов, разделенных точками. Рассмотрим примеры, иллюстрирующие эти правила.

Предположим, что в контейнерном объекте CLASS (см. рис. 9.6) создан объект пользователя (User) с именем USR. В табл. 9.13 показаны различные способы кодирования имени регистрации в команде LOGIN.

Таблица 9.13. **Примеры кодирования имени регистрации в команде LOGIN**

Имя регистрации в LOGIN	Имя контекста (Name Context) PC	Полный путь поиска объекта User (полное имя объекта) в дереве NDS (от оконечного объекта к корневому)
.USR.CLASS.MSTU (или .CN=USR.OU=CLASS.O=MSTU)	Произвольное	USR.CLASS.MSTU
USR (или CN=USR)	CLASS.MSTU	USR.CLASS.MSTU
USR.CLASS (или CN=USR.OU=CLASS)	MSTU	USR.CLASS.MSTU
ADMIN. (или CN= ADMIN.)	CLASS.MSTU	ADMIN.MSTU

Новые по сравнению с NetWare 3.x возможности администрирования на уровне NDS

В табл. 9.14 перечислены новые по сравнению с NetWare 3.x возможности администрирования на уровне NDS NetWare 4.x/5.x.

NetWare 4.x/5.x предлагает средства копирования дерева NDS или его частей (*разделов*) на разные файловые серверы. Разделы являются логическими единицами. Физические копии разделов называют *репликами*. Использование копий разделов позволяет повысить:

надежность системы, так как при выходе из строя какого-либо сервера пользователь может подключиться к сети, используя копию раздела NDS на другом сервере;

производительность системы, так как в распределенных сетях требуемые части дерева NDS можно хранить на серверах, которые располагаются ближе к пользователю.

С помощью утилиты NDS Manager Utility можно выполнить следующие операции над разделами и репликами:

- создать раздел,
- объединить два раздела в один,
- создать реплику,
- изменить тип реплики,
- удалить реплику.

Таблица 9.14. **Сравнение возможностей администрирования NetWare 3.x и NetWare 4.x/5.x**

NetWare 3.x	NetWare 4.x/5.x
На каждом сервере создаются разные базы данных Bindery	База данных NDS – это база данных всей сети. На раз-ных серверах можно хранить копии базы данных NDS или ее частей (разделов)
Администрирование выполняется с помощью разных текстовых утилит (SYSCON, FILER, SALVAGE и т. д.)	Основное средство администрирования – это графическая утилита (Windows-программа) NetWare Administrator, позволяющая выполнять все основные функции администратора: – защиту объектов дерева NDS и их свойств (опекуны, IRF), – защиту файловой системы каждого сервера, который определен в дереве NDS (опекуны, IRF, атрибуты), – описание объектов печати, – описание пользователей, групп, менеджеров, операторов, – описание сценариев подключения всех типов, – управление печатью и др.

Создание раздела. В процессе инсталляции первого сервера сети автоматически создается первый раздел NDS, началом которого служит объект [Root] (корень дерева), и Master-реплика этого раздела.

Для создания нового раздела вручную необходимо указать на контейнер – корень будущего раздела. При этом автоматически создается Master-реплика нового раздела. Она сохраняется на сервере, который хранит Master-реплику родительского раздела (для первого раздела, создаваемого вручную, родительским разделом будет раздел [Root]). Кроме того, если существуют другие серверы, хранящие Read/Write(Read-Only)-реплики этого же родительского раздела, то на них автоматически создаются Read/Write(Read-Only)-реплики создаваемого раздела.

Создаваемые разделы всегда покрывают все дерево NDS и никогда не перекрываются между собой (рис. 9.11).

Объединение двух разделов в один. Можно добиться соединения родительского и дочернего разделов, указав вручную на дочерний раздел. Это следует делать, если разделы содержат логически связанную информацию или их реплики должны быть объединены и располагаться на одних и тех же серверах.

Создание реплики (новой копии раздела). Для создания новой реплики необходимо указать на контейнер (корень раздела), выбрать сервер, на котором необходимо создать реплику, и определить тип этой реплики (Read/Write или Read-Only).

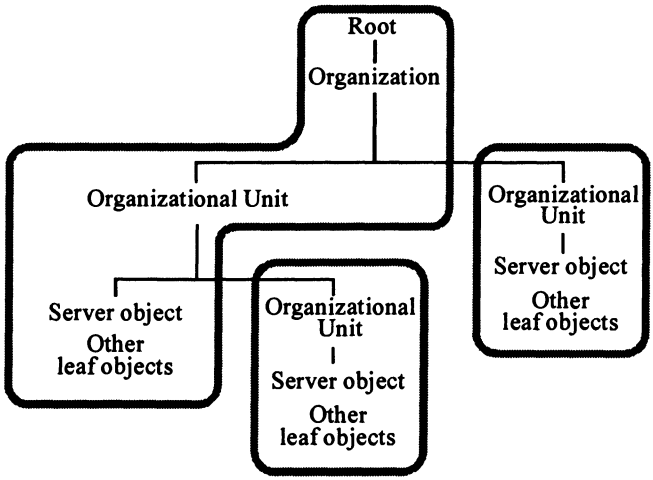


Рис. 9.11. Разделы NDS

При этом следует помнить, что каждый раздел может иметь только одну Master-реплику (все остальные реплики должны быть Read/Write или Read-Only). На сервере можно хранить только одну реплику раздела.

Изменение типа реплики. При изменении типа реплики следует учитывать, что

изменение Read/Write(Read-Only)-реплики на Master-реплику автоматически изменяет тип старой Master-реплики на Read/Write,

можно изменить тип Read/Write-реплики на Read-Only и наоборот без влияния на другие реплики этого раздела.

Удаление реплики. При удалении реплики необходимо помнить следующее:

нельзя удалить Master-реплику; при необходимости следует изменить тип какой-либо Read/Write(Read-Only)-реплики на Master, при этом старая Master-реплика автоматически преобразуется в Read/Write-реплику, которую уже можно удалить,

для обеспечения устойчивости к отказам следует иметь, как минимум, две реплики для каждого раздела.

Средства миграции с Bindery в NDS

Существует несколько способов преобразования сервера NetWare 3.x в сервер NetWare 4.x/5.x:

- с помощью программы инсталляции NetWare 4.x/5.x,
- с помощью утилиты MIGRATE (режим миграции данных с сервера NetWare 3.x на сервер NetWare 4.x/5.x),
- с помощью утилиты MIGRATE (режим миграции данных на то же самое устройство),
- с помощью средства DS STANDART фирмы Preferred Systems.

Рассмотрим первые три способа, предлагаемые фирмой Novell.

Использование программы инсталляции NetWare 4.x/5.x. На сервере NetWare 3.x необходимо запустить файл INSTALL с устройства CD-ROM (с ОС NetWare 4.x/5.x) и выбрать режим обновления версии (Upgrade). Далее требуется следовать инструкциям утилиты INSTALL.

Эта процедура во многом похожа на процедуру инсталляции NetWare 4.x/5.x (§ 9.1). Только здесь не нужно создавать раздел и тома NetWare, так как используется старая разметка диска.

Использование утилиты MIGRATE *в режиме миграции данных с сервера* NetWare 3.x *на сервер* NetWare 4.x /5.x (рис. 9.12). Рассмотрим процесс миграции данных с сервера NetWare 3.x на сервер NetWare 4.x.

Первоначально необходимо переписать директорию MIGRATE с CD-ROM (с ОС NetWare 4.x) на локальный диск PC. Далее следует выполнить следующие шаги:

1. Запустить утилиту MIGRATE на PC.
2. Выбрать тип миграции «Across-the-Wire Migration».
3. Выбрать тип исходного сервера (NetWare 3.x) и сервера-получателя (NetWare 4.x).
4. Выбрать рабочую директорию на PC для хранения данных Bindery.
5. Выбрать исходный сервер, информацию в базе данных Bindery для передачи, тома сервера с передаваемыми данными.
6. Выбрать сервер-получатель, тома для приема данных, опцию назначения паролей.

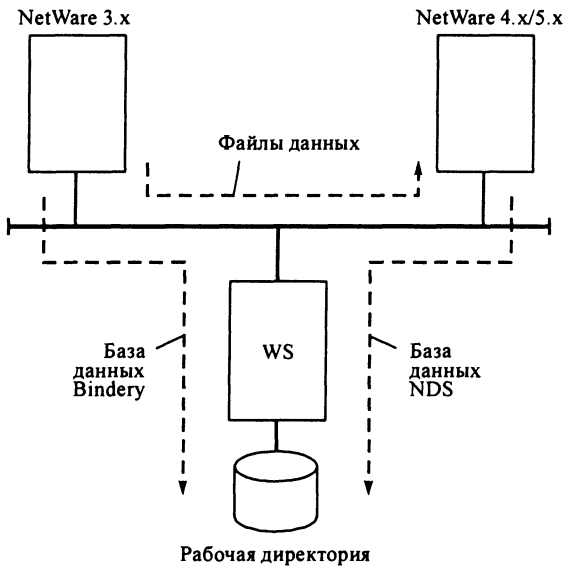


Рис. 9.12. Миграция данных с сервера NetWare 3.x на сервер NetWare 4.x/5.x

При выборе опции «Assign Random Passwords» система генерирует случайные пароли пользователей и сохраняет их в файле SYS:SYSTEM\NEW.PWD сервера NetWare 4.x. При подключении к серверу NetWare 4.x клиент должен использовать соответствующий ему пароль из этого файла.

При выборе опции «Assign no Passwords» пользователь не должен вводить пароль при первом подключении к NetWare 4.x.

7. В меню, которое появляется после нажатия клавиши F10, следует выбрать пункт «Start the Migration».

Утилита MIGRATE перемещает файлы с сервера NetWare 3.x на сервер NetWare 4.x. Информация базы данных Bindery сохраняется в рабочей директории PC, а затем перемещается в базу данных NDS NetWare 4.x.

8. Выйти из утилиты.

Желательно, чтобы структуры файловых систем NetWare 3.x и 4.x совпадали. Это уменьшает число переспросов утилиты о том, куда копировать файлы.

Использование утилиты MIGRATE в режиме миграции данных на то же устройство (рис. 9.13). Как и в предыдущем режиме, необходимо скопировать директорию MIGRATE с дистрибутивного CD-ROM NetWare 4.x на PC и выполнить следующие шаги:

1. Сохранить файлы NetWare 3.x на стриммере PC.
2. Запустить утилиту MIGRATE на PC.
3. Выбрать тип миграции «Same-Server Migration».
4. Выбрать тип исходного сервера (NetWare 3.x) и сервера-получателя (NetWare 4.x).
5. Выбрать рабочую директорию на PC для хранения данных Bindery.
6. Выбрать исходный сервер, информацию в базе данных Bindery для передачи, тома сервера для сохранения таблиц DET и FAT.
7. В меню, которое появляется после нажатия клавиши F10, выбрать пункт «Migrate to the Working Directory». В результате база данных Bindery перемещается в рабочую директорию на PC.

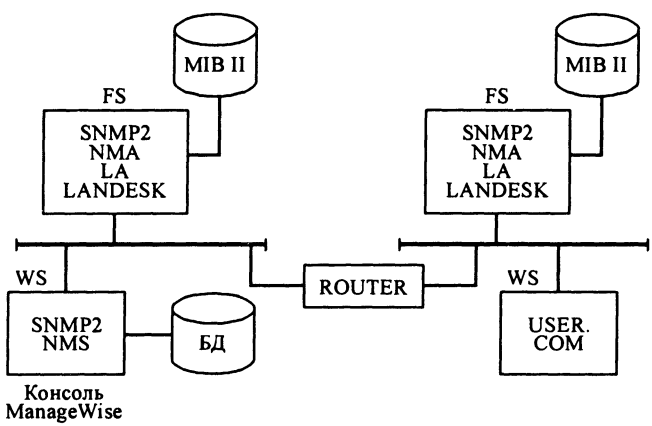


Рис. 9.13. Миграция данных на то же устройство

8. Выйти из утилиты.

9. Проинсталлировать файловый сервер NetWare 4.x на том же устройстве, где размещалась ОС NetWare 3.x.

10. Восстановить на сервере NetWare 4.x файлы со стриммера.

11. Запустить утилиту MIGRATE и повторить п. 3, 4, 5.

12. Выбрать сервер-получатель, тома сервера для восстановления таблиц DET и FAT, опцию назначения паролей.

13. В меню, которое появляется после нажатия клавиши F10, выбрать пункт «Migrate from Working Directory». В результате информация из рабочей директории перемещается в базу данных NDS.

14. Выйти из утилиты.

Этот режим используют, если требуется существенно перестроить файловую систему сервера.

9.4. Оперативное управление

Обзор команд оперативного управления

В табл. 9.15 перечислены некоторые команды оперативного управления, которые можно вводить с консоли файлового сервера NetWare.

Таблица 9.15. **Команды оперативного управления**

Команда	Описание
ADD NAME SPACE имя том	Создать на томе новое пространство имен
BIND протокол [TO] имя_драйвера | имя_конфигурации_карты [[параметры_драйвера]] NET=номер_сети	Привязать протокол к драйверу сетевого адаптера
BROADCAST "сообщение" [[TO] имя | соединение ...]	Передать сообщение всем или нескольким пользователям, которые зарегистрированы в сети
CONFIG	Отобразить характеристики файлового сервера, указанные в файле AUTOEXEC.NCF
DISPLAY NETWORKS	Отобразить сведения о топологии сети
DISPLAY SERVERS	Отобразить сведения об имеющихся в сети серверах, мостах и маршрутизаторах
DISABLE LOGIN	Запретить новым клиентам подключаться к серверу
DISMOUNT имя_тома	Размонтировать том сервера
ENABLE LOGIN	Разрешить новым клиентам подключаться к серверу
LOAD [путь]имя_модуля[параметры_ модуля]	Загрузить и выполнить NLM-модуль
MEMORY	Показать объем оперативной памяти сервера

Окончание табл. 9.15

Команда	Описание
MODULES	Посмотреть список загруженных NLM-модулей
MOUNT имя_тома | ALL	Смонтировать один или все тома сервера
NAME	Отобразить имя файлового сервера
OFF или CLS	Очистить экран консоли файлового сервера
REMOVE DOS	Выгрузить DOS из ОП сервера. Освободившаяся память присоединяется к кэш-буферу (3.x, 4.x)
RESTART SERVER [параметры]	Перезапустить сервер
SEARCH	Отобразить пути поиска NLM-модулей
SEARCH ADD номер путь	Добавить путь поиска NLM-модулей
SEARCH DEL номер	Удалить путь поиска NLM-модулей
SECURE CONSOLE	Выгрузить DOS из ОП сервера и разрешить чтение NLM-модулей только из системной директории SYS:SYSTEM
SET	Посмотреть или изменить SET-параметры ОС
SET TIME [месяц/день/год][час:минуты:секунды]	Установить системную дату и время
TIME	Посмотреть системную дату и время
VERSION	Отобразить версию и характеристики лицензии NetWare
VOLUMES	Отобразить список томов, смонтированных на файловом сервере
UNBIND протокол [FROM] имя_драйвера [параметры_драйвера]	Отсоединить протокол от драйвера сетевого адаптера
UNLOAD имя_модуля	Выгрузить NLM-модуль из ОП
DOWN	Завершить работу ОС NetWare
EXIT	Выйти в DOS файлового сервера после выполнения команды DOWN. Если до этого DOS был выгружен (см. команды SECURE CONSOLE и REMOVE DOS), то выполняется перезагрузка NetWare (3.x, 4.x)

Наблюдение и контроль состояния системы

Мониторинг сети – это оценка ее характеристик производительности и надежности. Для мониторинга сети NetWare, в частности, можно использовать следующие средства:

- NLM-модуль MONITOR.NLM,

• продукт NetWare Management System (NMS),
• продукт ManageWise.

Модуль MONITOR.NLM является утилитой файлового сервера NetWare и позволяет анализировать следующие важные характеристики производительности и надежности системы:

1) по каждому пользователю:
   – количество запросов к файловому серверу, пришедших от PC за время соединения,
   – объем данных, прочитанных и записанных PC на диск сервера за время соединения;
2) по каждому жесткому диску сервера:
   – состояние средства переадресации дефектных блоков (Hot Fix),
   – число переадресованных дефектных блоков,
   – состояние режима проверки чтением после записи на диск;
3) по каждому сетевому адаптеру сервера:
   – общее число отправленных и принятых пакетов,
   – число пакетов, при приеме которых не было свободных блоков ECB,
   – количество кадров, которые были приняты с ошибками и др.;
4) по оперативной памяти сервера:
   – размер пула выделенной памяти,
   – распределение пула выделенной памяти между NLM-модулями (для NetWare 4.x/5.x),
   – размер кэш-буфера,
   – характеристики использования кэш-буфера (для NetWare 4.x/5.x),
   – размер области перемещаемой памяти,
   – размер области, занимаемой NLM-модулями (код и данные);
5) по процессору сервера:
   – загрузка процессора за последнюю секунду,
   – загрузка процессора NLM-модулями и программами обработки прерываний;
6) сведения о ядре NetWare (5.x):
   – информация о процессорах,
   – информация о приложениях и их нитях.

Продукт NMS разработан фирмой Novell, а средство ManageWise является совместной разработкой Novell и Intel. ManageWise включает в себя продукты NMS (Novell) и Landesk (Intel).

Средство ManageWise включает следующие компоненты (рис. 9.14):

NMS:

• LANalyzer Agent (LA) – агент сервера для сбора информации о сегменте сети;
• Network Management Agent (NMA) – агент сервера для сбора информации о файловом сервере;
• NetWare Management System (NMS) – консоль ManageWise, выполняемая под Windows.

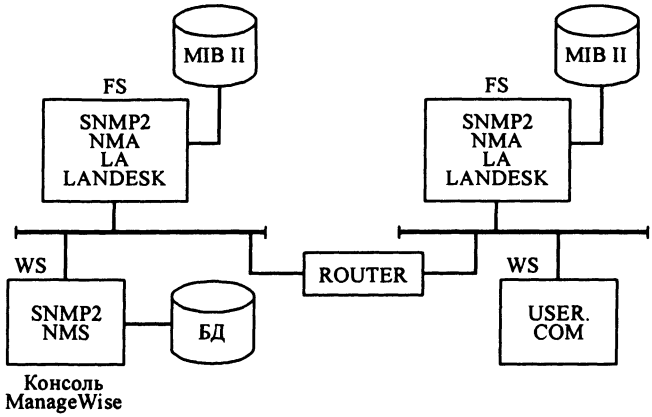


Рис. 9.14. Компоненты ManageWise

LANDESK:

- средство удаленного управления PC, устанавливаемое на сервере,
- программа-агент, устанавливаемая на PC (USER.COM),
- средство поиска вирусов в сети, устанавливаемое на сервере.

При инсталляции ManageWise устанавливаются:

программы-агенты на файловых серверах, для которых необходимо осуществлять мониторинг,

программы-агенты на PC, которыми надо управлять,

Windows-программы на станции администратора для реализации функций мониторинга и управления (консоль ManageWise).

После запуска под Windows консоли ManageWise (NMS) она взаимодействует с программами-агентами с помощью трех основных команд:

GET – опросить базу данных MIB (Management Information Base). Эта база содержит информацию о форматах данных, с помощью которых NMS и программы-агенты могут обмениваться между собой. NMS транслирует сведения из этой базы и запоминает их в собственной базе данных на станции администратора;

TRAP – получить данные от программы-агента. Как только администратор выбирает требуемую характеристику, NMS формирует команду TRAP для соответствующего агента. После этого программа-агент начинает пересылать на станцию администратора требуемые сведения. NMS отображает их в соответствующем окне;

SET – установить параметр. С помощью этой команды NMS управляет (через программы-агенты) состоянием процессора, PC и др.

LA-агент перехватывает пакеты, поступающие на сервер, накапливает статистику о функционировании сегмента и пересылает ее на станцию администратора. NMA-агентом накапливается статистика о работе файлового сервера.

Связь между агентами и базой данных MIB, а также между агентами и консолью ManageWise осуществляется с помощью протокола SNMP (Simple Network Management Protocol). SNMP является стандартным протоколом для разработки различных систем администрирования сетей. Он используется в системах администрирования OpenView (HP), NetView/6000 (IBM), Norton Administrator for Networks (Symantec).

С помощью консоли ManageWise можно выполнять следующие функции.

1. Просматривать графики и анализировать информацию:

а) по сегментам:

производительность сегмента (пакет/с),

загрузку сегмента,

количество ошибок в 1 с,

текущий трафик (пакет/с) и т. д.,

б) по файловым серверам (где установлены агенты):

загрузку процессора,

статистику о работе контроллера жесткого диска,

статистику о работе сетевого адаптера,

информацию о NLM-модулях,

в) по сети в целом:

топологию сети,

индикаторы и сообщения об ошибках и предупреждениях (например, о превышении установленного порога для кэш-буфера).

2. Вырабатывать управляющие воздействия

а) для настройки параметров файловых серверов (средствами удаленной консоли),

б) для настройки концентраторов (включить, выключить порты и т. д.),

в) для настройки PC (LANDESK).

Управление сетевой печатью

Управляемые компоненты сетевой печати перечислены в табл. 9.16.

Таблица 9.16. **Управляемые компоненты сетевой печати**

Управляемые компоненты	Управляет	Утилиты управления
Очереди заданий на печать	Администратор сети	PCONSOLE.EXE (3.x, 4.x), NetWare Administrator (4.x, 5.x)
	Оператор очереди	PCONSOLE.EXE (3.x, 4.x), NetWare Administrator (5.x)
Принтеры	Администратор сети	PCONSOLE.EXE (3.x, 4.x), NetWare Administrator (4.x, 5.x), PSERVER.NLM (4.x, 5.x)
	Оператор сервера печати	PCONSOLE.EXE (3.x, 4.x), NetWare Administrator (5.x), PSERVER.NLM (4.x, 5.x)

Управление очередью заданий на печать включает

• управление заданиями в очереди:

задержку задания в очереди,

изменение номера обслуживания задания в очереди (т. е. изменение приоритета задания в очереди),

• управление состоянием очереди:

возможность постановки клиентом задания в очередь,

возможность обслуживания очереди сервером печати,

возможность подключения нового сервера печати к очереди.

Управление принтерами включает:

• изменение приоритета очереди, обслуживаемой принтером,

• изменение списка пользователей, оповещаемых о проблемах, возникающих при работе принтера (отсутствие бумаги и т. д.),

• управление состоянием принтера:

останов принтера,

пауза принтера,

старт принтера,

сброс задания,

прогон листа,

печать строки из звездочек.

9.5. Разработка приложений для NetWare

Средства разработки приложений

Многие годы компания Novell предлагала программистам на языке C API-программы для разработки загружаемых модулей NetWare (NetWare Loadable Module – NLM), выполняемых на сервере. Начиная с IntranetWare (NetWare 4.11), компания стала включать в свои ОС язык NetBasic с целью предоставления программистам средства, подобного Visual Basic, для написания серверных программ. Но все-таки NetWare уступает Windows NT в части предоставления программных средств для разработки сервера приложений.

Чтобы преодолеть отставание в области серверов приложений компания Novell включила в NetWare 5 средства поддержки Java на сервере. Идея состоит в том, что серверы NetWare должны стать быстрыми, защищенными платформами с интегрированными каталогами для выполнения программ на языке Java. Если Java будет и дальше развиваться как инструментарий разработки корпоративного уровня, если большое количество приложений и компонентов на языке Java появятся на рынке, то стратегия Novell окажется очень перспективной. Усилия, предпринимаемые Enterprise Java Beans компании Sun и IBM, могут дать инструментарий и импульс для создания серверных приложений на Java.

Ниже рассмотрены правила написания NLM-модулей на языке C.

Структура и правила написания NLM-модулей на языке С

Основным средством разработки NLM-модулей, загружаемых на файловом сервере, является компилятор Novell/Watcom C Network Compiler/386. Это связано с тем, что только этот С-компилятор имеет в своем составе редактор NLMLINK, позволяющий компоновать NLM-модули. Компоновщик WLINK, входящий в состав компилятора Novell/Watcom Network C for DOS (средство разработки приложений для DOS-рабочей станции), также позволяет создавать NLM-модули.

Для разработки NLM-модуля необходимо выполнить следующие действия.

1. Разработать тексты С-программ. Это обычные программы на языке С, где используются обращения к службам NetWare (примитивы NetWare для NLM-модулей).
2. Разработать make-файл (рис. 9.15).
3. Запустить утилиту WMAKE для создания NLM-модуля:

WMAKE –f make-файл имя.nlm

В make-файле описываются имена компилятора и компоновщика, а также состав каждого obj-файла и nlm-файла. Если С-файл был изменен, то после запуска утилиты WMAKE выполняется компиляция этого файла и перекомпоновка obj- и nlm- файла.

Файл определений, имя которого указывается в make-файле, имеет расширение DEF и содержит информацию для компоновщика. В табл. 9.17 приведены ключевые слова, используемые в файле определений.

В файле определений *.DEF обязательными являются ключевые слова DESCRIPTION, INPUT, OUTPUT, IMPORT.

Для разработки прикладных программ, которые выполняются на PC, можно использовать С-компиляторы фирм Watcom, Microsoft, Borland и Lattice. Для разработки сетевых приложений необходимо дополнительно приобрести интерфейс NetWare C Interface, включающий примитивы NetWare. Интерфейс поставляется в виде библиотек для С-компиляторов, перечисленных выше. Компилятор Novell/Watcom Network C for DOS уже имеет в своем составе соответствующую библиотеку из С интерфейса. С помощью этого компилятора можно помимо DOS-приложений создавать и NLM-модули для файлового сервера.

```
# Описание переменных make-файла, используемых при
# задании параметров программ
перменная=значение
перменная=значение
...
# Определение имени компилятора
.C.OBJ:
wc1386 параметры
# Описание состава obj-файлов
имя.obj: имя.c имя.c ... имя.h
имя.obj: имя.c имя.c ... имя.h
...
# Описание состава NLM-модуля
имя.nlm: имя.obj имя.obj ... файл_определений.def
# Описание имени компоновщика
nlmlink файл_определений
```

Рис. 9.15. Описание make-файла, используемого при создании NLM-модуля

Таблица 9.17. **Ключевые слова файла определений**

Ключевое слово	Описание
DESCRIPTOR	Строка до 127 символов, заключенная в двойные кавычки. Эта строка будет отображаться на консоли при выполнении команды файлового сервера MODULES
INPUT	Список модулей и файлов для компоновки. Может содержать имена требуемых obj-файлов и файлов со списками obj-файлов. Файл со списком обозначается как @имя_файла. Каждая строка файла должна начинаться с пробела. В этом списке должен быть указан путь к файлу PRELUDE.OBJ. Модуль PRELUDE устанавливает среду NLM-программы с помощью – распределения и инициализации внутренних структур данных; – создания экрана для использования его NLM-программой; – разборки параметров командной строки для передачи их в процедуру main( ) как аргументов вида argc и argv; – старта новой нити для выполнения процедуры main( ). Пример: INPUT NW3NLM E:\WC386\PRELUDE @E:\WC386\A1.MOD
OUTPUT	Имя NLM-модуля (без расширения)
IMPORT	Список функций, доступных в других NLM-модулях. Может содержать имена функций и файлов со списками функций. Файл со списком обозначается как @имя_файла. Каждая строка файла должна начинаться с пробела. Перед загрузкой этого NLM-модуля должны быть загружены NLM-модули, содержащие функции, которые перечислены в списке IMPORT. Таким образом, NLM-модули могут выступать в качестве динамических библиотек. Они загружаются один раз и все NLM-модули, запускаемые в дальнейшем, могут использовать функции, которые описаны в списке EXPORT (см. ниже) этих модулей-библиотек. В списке IMPORT следует указывать файл @CLIB.IMP, объявляющий процедуры из стандартной С-библиотеки. Если используется арифметика с плавающей точкой, то необходимо указать файл @MATHLIB.IMP. Пример:IMPORT @E:\WC386\CLIB.IMP @E:\WC386\MATHLIB.IMP
SCREENNAME	В двойных кавычках указывается имя экрана, автоматически создаваемого при загрузке NLM-модуля. Если имя не указано, то в качестве идентификатора экрана используется имя модуля
THREADNAME	Строка до 5 символов в двойных кавычках – шаблон для генерации имен новых нитей, создаваемых в загружаемом NLM-модуле. При формировании имени нити к шаблону добавляются цифры (по возрастанию, начиная с 0). Если имя не указано, то используются первые 5 символов имени модуля

Окончание табл. 9.17

Ключевое слово	Описание
EXPORT	Список имен функций, доступных другим NLM-модулям. В списке можно использовать имя файла (@имя_файла). Каждая строка файла должна начинаться с пробела. Все NLM-модули, загружаемые в дальнейшем, могут обращаться к этим функциям в случае, если они описаны в списках IMPORT этих модулей
MAP	Запрашивает создание MAP-файла. Если имя не указано, будет использоваться имя из директивы OUTPUT
DEBUG	Запрашивает, чтобы компоновщик включал дополнительную информацию для отладчика
CHECK	Указывает функцию, вызываемую перед выгрузкой NLM-модуля
STACK	Размер стека, выделяемого при инициализации процесса, связанного с NLM-модулем. По умолчанию – 8192 б
MULTIPLE	Разрешает загрузку нескольких копий NLM-модулей. Иначе при попытке загрузки более одной копии на консоль файлового сервера будет выдано сообщение об ошибке

Тексты С-программ – обычные программы на языке С (или С++), где используются обращения к службам NetWare, т. е. к примитивам NetWare для PC.

Организация API-интерфейса для разработки программ на языке С

Для обращений к службам NetWare из NLM-модулей и программ, которые выполняются на PC, используются так называемые примитивы (функции). Эти примитивы образуют API-интерфейс связи прикладных программ с NetWare.

Часто имена этих примитивов совпадают для NLM-модулей и программ PC. Отличие заключается в том, что примитивы, используемые в NLM-модуле, хранятся в других NLM-модулях, выступающих в роли общих библиотек (CLIB.NLM, MATHLIB.NLM и т. д.). Эти функции вызываются динамически, а не дублируются в разных NLM-модулях. Примитивы, используемые в прикладной программе PC редактируются (линкуются) с основной программой, являются составной частью исполняемого модуля и дублируются в разных программах.

В табл. 9.18 перечислены названия заголовочных файлов (h-файлов) различных служб NetWare 3.x. В h-файлах хранятся константы и прототипы примитивов. Следует отметить, что соответствующие службы имеются и в NetWare 4.x. Большинство функций NetWare 4.x выполняется в режиме Bindery, т. е. так же, как и в NetWare 3.x.

Таблица 9.18. **Заголовочные файлы служб NetWare 3.x**

Служба NetWare	h-файл для создания NLM-модуля	h-файл для создания программы PC
Служба протоколов IPX и SPX	NWIPXSPX.H	NXT.H
Служба учета счетов	NWACCTNG.H	NWACCT.H
Служба базы объектов	NWBINDRY.H	NWBINDRY.H
Служба соединений	NWCONN.H	NWCONN.H
Служба среды файлового сервера	NWENVRN.H	NWCONSOL.H
Служба каталогов	DIRECT.H, NWDIR.H	NWDIR.H
Служба файлов	NWFILE.H	NWFILE.H, NTT.H
Служба сообщений	NWMSG.H	NWMSG.H
Служба печати	–	NWPRINT.H
Служба сервера печати	–	NPT.H
Служба синхронизации	NWSYNC.H	NWSYNC.H
Служба отслеживания транзакций	NWTTS.H	NWTTS.H
Служба среды рабочей станции	–	NWWRKENV.H

API-интерфейсы для NLM-модулей и программ, выполняемых на PC, включают более 600 примитивов.

10. ВЫБОР АРХИТЕКТУРЫ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ

Глава посвящена вопросам практики построения сетей ЭВМ. Приведена общая схема взаимодействия локальных, городских и глобальных вычислительных сетей. Выбор локальной вычислительной сети описан в контексте сравнения различных технологий локальных сетей: 10-мегабитные сети Ethernet и проблемы перехода к 100-мегабитным сетям, Fast Ethernet и 100VG-AnyLan, Token Ring и 10-мегабитные сети Ethernet. Выбор магистрали для объединения локальных сетей в черте города и магистрали WAN для объединения сетей в разных городах описан в контексте сравнения таких технологий, как FDDI и ATM для первого случая и X.25, Frame Relay и ISDN для второго. Завершает главу описание тенденций развития технических средств для распределенной обработки

10.1. Общая схема взаимодействия локальных, городских и глобальных вычислительных сетей

Разработка систем распределенной обработки данных предполагает разделение данных по различным, возможно, удаленным узлам. Часто архитектура распределенной системы включает три уровня:

1. Локальные сети (LAN – Local Area Networks) – сети компьютеров, сосредоточенные на небольшой территории. Протяженность сети составляет 1...2 км. Как правило, LAN-сеть располагается в одном или нескольких соседних зданиях и принадлежит одной организации.

2. Городские сети, сети метрополий (MAN – Metropolitan Area Networks) – сети, объединяющие локальные сети в черте города. Протяженность сети – 100...200 км.

3. Глобальные сети (WAN – Wide Area Networks) объединяют территориально рассредоточенные сети LAN и MAN, а также удаленные компьютеры. WAN-сети охватывают практически все страны мира.

На рис. 10.1 представлена схема взаимосвязи сетей LAN, MAN и WAN.

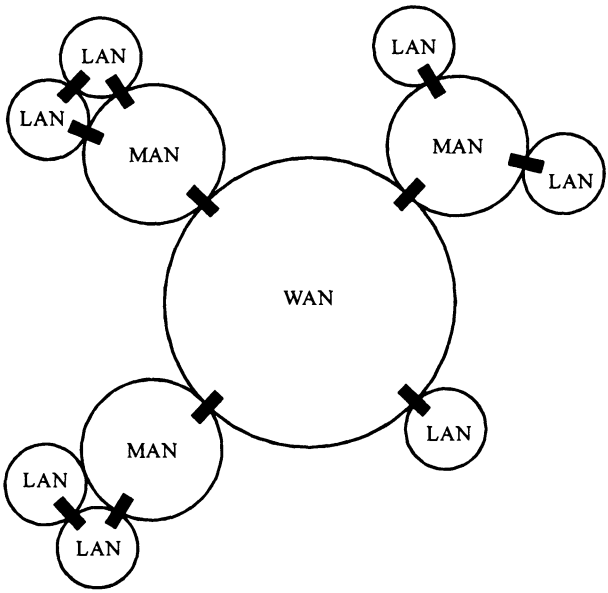


Рис. 10.1. Схема взаимосвязи различных типов сетей

Сети объединены между собой маршрутизаторами и коммутаторами. Маршрутизатор, в отличие от коммутатора (многопортового моста), выбирает оптимальный путь передачи данных.

Ниже рассмотрены различные решения для указанных выше типов сетей.

10.2. Выбор локальной вычислительной сети

10-мегабитные сети Ethernet: варианты, преимущества, недостатки

В настоящее время при построении 10-мегабитных локальных сетей Ethernet используют в основном:

- сети на тонком коаксиальном кабеле;
- сети на концентраторах;
- сети на коммутаторах.

Рассмотрим каждое из этих решений подробнее.

Сети на тонком коаксиальном кабеле. На рис. 10.2 показана типовая структура сети на тонком коаксиальном кабеле, в которой две сети связываются между собой через сервер (S0). Это возможно, так как практически любая сетевая ОС имеет в своем составе внутренний маршрутизатор.

Для соединения станций используют кабель RG58. Его разрезают на отрезки. К каждому концу отрезка прикрепляют BNC-коннектор (обжимается), который вставляется в Т-коннектор. А Т-коннектор, в свою очередь, соединен с сетевым адаптером (CA) станции. Сетевой адаптер представляет собой плату, которая вставляется в расширительный слот PC (WS) или сервера.

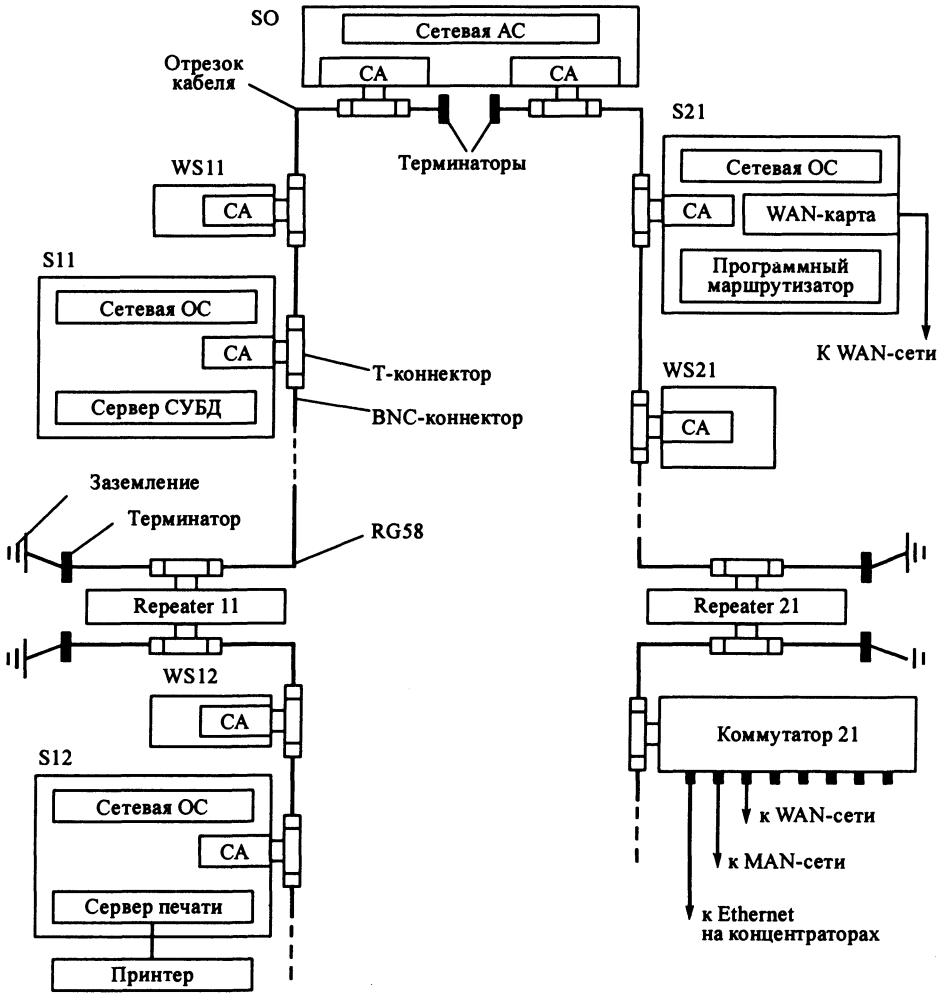


Рис. 10.2. Типовая структура сети на тонком коаксиальном кабеле

Сеть может состоять из нескольких сегментов. На двух концах сегмента устанавливают терминаторы, один из которых заземляется. Терминатор представляет собой заглушку с омическим сопротивлением 50 Ом, соединяющим оплетку с жилой коаксиального кабеля. Терминаторы необходимы, чтобы гасить сигналы при передаче пакетов (точнее кадров) по сети (т. е. чтобы не было отражения).

Сегменты сети соединены повторителями (repeater), которые служат усилителями сигналов. Повторители не выполняют «развязку» сегментов, т. е. по

сети не могут одновременно передаваться пакеты между несколькими парами станций. Например, в один и тот же момент времени по сети нельзя передавать данные между станциями WS11 и S11, а также между станциями WS12 и S12. Но сервер S0, дополнительно выполняющий роль маршрутизатора, выполняет «развязку» сетей, т. е. по сети одновременно могут передаваться кадры между PC WS11 и сервером S11, а также между WS21 и S21.

В табл. 10.1 перечислены некоторые важные ограничения для локальных сетей на тонком коаксиальном кабеле.

Таблица 10.1. **Некоторые ограничения для сетей на тонком коаксиальном кабеле**

Параметр сети	Значение
Длина отрезка кабеля	Не менее 2,5 м
Длина сегмента	Не более 185 м
Количество станций в одном сегменте	Не более 30
Количество сегментов в одной сети	Не более 5
Общее число станций в одной сети	Не более 90

К сегментам сети подключают PC, серверы, коммутаторы, аппаратные маршрутизаторы и другие специфические устройства (специальные сетевые принтеры, факс-серверы и др.). На PC устанавливают сетевое ПО и приложения для работы с базами данных. На серверах инсталлируют сетевые ОС (NetWare, Windows NT, Unix), а также устанавливают дополнительные продукты:

- программный маршрутизатор для обеспечения доступа клиентов к WAN-сети;
- серверы СУБД для организации работы клиентов в архитектуре клиент/сервер;
- серверы печати (они входят в состав сетевых ОС) для организации доступа всех (или части) клиентов сети к небольшому числу принтеров;
- другие продукты.

За рубежом сети на тонком коаксиальном кабеле практически не используют. В России они достаточно распространены, хотя в настоящее время все чаще устанавливают сети на витой паре. Для объединения старых сетей (на коаксиале) и новых сетей (на витой паре) используют концентраторы и коммутаторы (см. рис. 10.2). При этом коммутаторы могут обеспечить подключение к MAN-сетям (FDDI и ATM) и WAN-сетям (X,25, Frame relay, ISDN и др.).

В табл. 10.2 перечислены преимущества и недостатки сетей на тонком коаксиальном кабеле.

Таблица 10.2. **Преимущества и недостатки сетей на тонком коаксиальном кабеле**

Преимущества	Недостатки
Сети очень дешевые (сетевые адаптеры можно купить по цене меньше 30 долл.)	1. Скорость по сети ограничивается 10 Мбит/с (на самом деле практически меньше, все зависит от сетевых адаптеров)

Окончание табл. 10.2

Преимущества	Недостатки
Простота установки (для прокладки кабеля, установки сетевых адаптеров и выполнения соединений требуется минимальная квалификация персонала)	2. При обрыве или коротком замыкании на отрезке кабеля из строя выходит весь сегмент сети. Причем неисправность можно обнаружить или с помощью специального прибора (сканера), или вручную, последовательно перемещая терминатор от конца сегмента к его началу 3. При большом числе сегментов сеть перестает надежно работать. Это связано с тем, что заземления сегментов не могут быть идеальными, т. е. возникают отражения сигналов. При большом числе сегментов возникает интерференция (наложение) отраженных сигналов, что приводит к появлению помех 4. Станция работает в среднем со скоростью $10/N$ Мбит/с (N – число станций, подключенных к сети)

Сети на концентраторах (витой паре). На рис. 10.3 показана типовая структура сети на концентраторах.

Концентратор – это своего рода системный блок, имеющий слоты расширения, куда можно вставлять модули с портами. К этим портам можно подключать отдельные станции (PC, серверы и т. д.), концентраторы, коммутаторы и маршрутизаторы. Для их подключения используют, как правило, витую пару (10BaseT), реже – оптоволокно (10BaseFL). Сети на тонком коаксиальном кабеле подключают к порту 10Base2 концентратора.

В настоящее время многие фирмы выпускают самые разнообразные концентраторы: от простейших стоимостью до 200 долл. (FlexHub EHub-9 – с одним модулем на 8 портов 10BaseT и 1 порт 10Base2) до более сложных концентраторов, соединяемых в стек, стоимостью до 1000 долл. (FlexHub EHub-16 с 18 портами).

Простой концентратор выполняет функцию повторителя Ethernet. Можно соединить в каскад до четырех таких концентраторов. Следует различать каскадирование концентраторов от их соединения в стек. При каскадировании (см. рис. 10.3) концентраторы соединяют витой парой или оптоволокном и, естественно, их можно разнести в пространстве. При соединении концентраторов в стек они выступают как одно устройство с большим числом портов. Как правило, концентраторы устанавливают друг на друга в одном месте. Для их объединения используют специальные стековые порты.

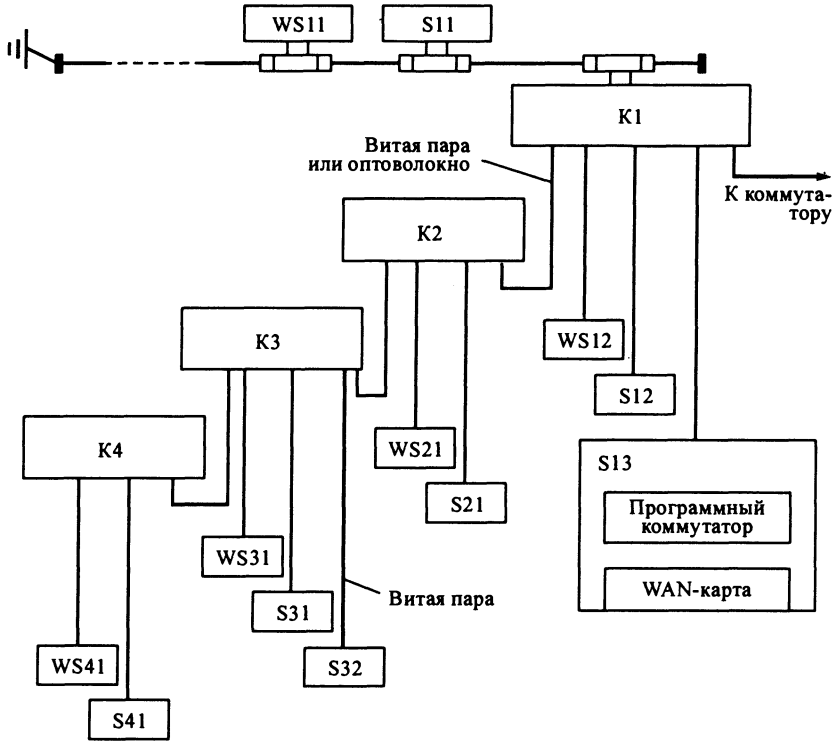


Рис. 10.3. Типовая структура на концентраторах

Концентраторы (К1 – К4 на рис. 10.3) не выполняют развязку сегментов сети, т. е. в один и тот же момент времени не могут передаваться пакеты, например, между PC WS12 и сервером S11, а также между WS31 и S31.

В табл. 10.3 перечислены некоторые ограничения для локальных сетей на концентраторах.

Таблица 10.3. **Некоторые ограничения для сетей на концентраторах**

Параметр сети	Значение
Длина неэкранированной витой пары (две пары)	Не более 100 м
Длина оптоволокна	Не более 2 км
Максимальное расстояние между двумя самыми удаленными станциями	2,5 км
Количество концентраторов в сегменте	Не более 4

В табл. 10.4 приведены основные преимущества и недостатки сетей на концентраторах.

Таблица 10.4. **Преимущества и недостатки сетей на концентраторах**

Преимущества	Недостатки
1. Если при прокладке использовалась витая пара, то достаточно просто можно осуществить переход на 100-мегабитные сети 100VG. Для этого следует заменить концентраторы и сетевые адаптеры (10-мегабитные на 100-мегабитные) 2. Если используется витая пара, то в сети нет заземления. Поэтому сеть работает надежнее 3. При обрыве или коротком замыкании витой пары из строя выходит одна станция 4. Простота установки 5. При переходе на 100-мегабитные сети средняя скорость PC в сети увеличивается до $100/N$ Мбит/с 6. При использовании оптоволокна кабель можно помещать в сильное магнитное поле, погружать в водоемы, вмерзать в лед	1. Несколько дороже сетей на коаксиальном кабеле (в основном на стоимость концентраторов). 2. Станция работает в среднем со скоростью $10/N$ Мбит/с (N – общее число станций, подключенных к концентраторам сети)

Сети на коммутаторах. Коммутаторы еще называют многопортовыми мостами. На рис. 10.4 показана типовая структура сети на коммутаторах.

К коммутатору можно подключать либо отдельные станции, либо сети (к портам, поддерживающим множество MAC-адресов). Коммутатор, в отличие

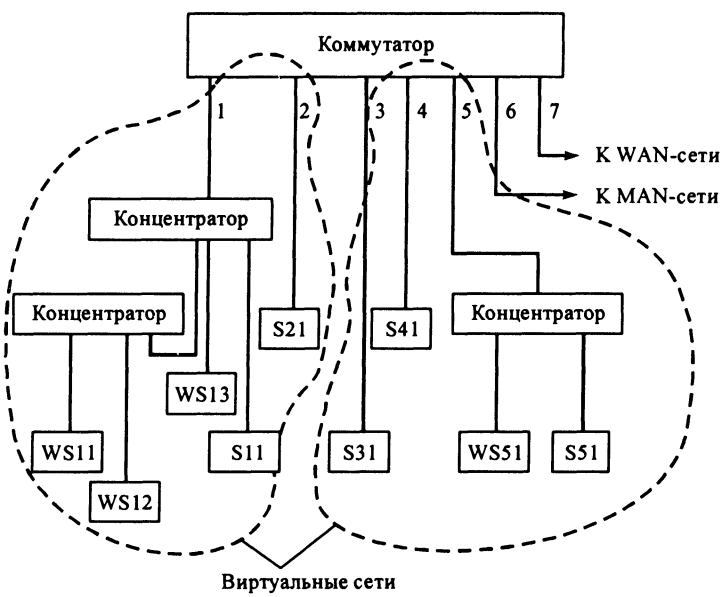


Рис. 10.4. Типовая структура сети на коммутаторах

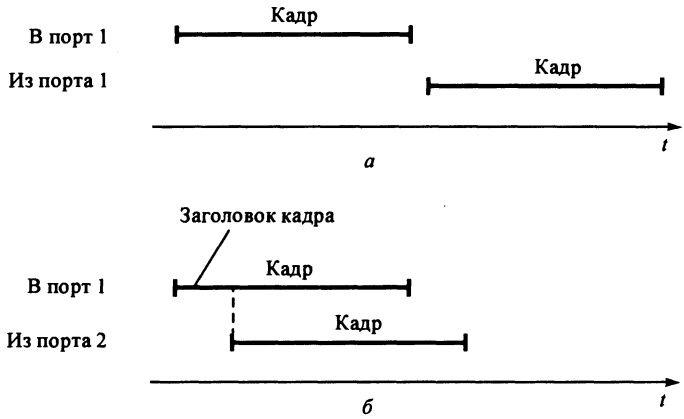


Рис. 10.5. Передача кадров с буферизацией (*а*) и без буферизации (*б*)

от концентратора, выполняет развязку портов, т. е. одновременно могут передаваться кадры, например, между PC WS11 и сервером S11, а также между WS51 и S51.

Коммутаторы характеризуются следующими особенностями функционирования:

- 10-мегабитный коммутатор поддерживает для каждого порта скорость 10 Мбит/с. Например, если к каждому порту коммутатора подключена только одна станция, то каждая станция может работать со скоростью 10 Мбит/с.
- Многие коммутаторы могут работать в двух режимах: с буферизацией и без буферизации (рис. 10.5).

При передаче кадров с буферизацией весь кадр сначала запоминается в буфере коммутатора, а затем передается в порт назначения. Этот режим целесообразно использовать для ненадежного оборудования. В этом случае при ошибке передачи кадра между станциями переспрос будет осуществляться на уровне «станция→коммутатор» или «коммутатор→станция». При передаче кадров без буферизации коммутатор принимает и анализирует заголовок кадра, а затем перенаправляет этот кадр в порт назначения. Здесь скорость передачи выше, но этот режим целесообразно использовать для надежного оборудования. При ошибке передачи кадра между станциями переспрос будет осуществляться на уровне «станция→коммутатор→станция».

- На коммутаторах можно создавать виртуальные сети. Во многих ОС серверы выполняют широковещательную рассылку пакетов с уведомлением других серверов о предоставляемых ими услугах (SAP-, RIP-пакеты и т. д.). Трафик этих пакетов в сети довольно большой. Но часто к коммутатору подключают логически изолированные группы станций (на рис. 10.4 они обведены контурами). Каждую из этих групп администратор сети может описать как виртуальную сеть. Далее он должен определить режимы фильтрации пакетов при их

передаче из одной виртуальной сети в другую. Например, можно запретить передачу широковещательных пакетов, в этом случае эти пакеты будут распространяться в рамках одной виртуальной сети. Маршрутизацию пакетов между виртуальными сетями выполняет коммутатор.

• Коммутатор может поддерживать сразу несколько процессов передачи. Например (см. рис. 10.4), он может одновременно передавать кадры из порта 1 в порт 2, из порта 3 в порт 4, из порта 5 в порт 6.

Конечно, коммутаторы имеют ощутимое преимущество перед концентраторами, но они в несколько раз дороже.

Проблемы перехода к 100-мегабитным сетям

В настоящее время для построения 100-мегабитных сетей используют две технологии: Fast Ethernet и 100VG-AnyLan. Рассмотрим каждый из этих вариантов.

Стандарт Fast Ethernet разработан в 1995 г. и получил название IEEE 802.3u. С легкой руки организации IEEE (Institute of Electrical and Electronic Engineers) Fast Ethernet именуется как 100BaseT. Это объясняется просто: 100BaseT является расширением стандарта 10BaseT с пропускной способностью 10 Мбит/с. Стандарт 100BaseT включает в себя протокол обработки множественного доступа с опознаванием несущей и обнаружением конфликтов CSMA/CD (Carrier Sense Multiple Access/Collision Detection). Fast Ethernet может работать с использованием различных кабелей (табл. 10.5).

Таблица 10.5. **Кабели, используемые в Fast Ethernet**

Обозначение	Описание	Длина до концентратора, м
100BaseTX	Две неэкранированные витые пары (UTP) категории 5 или две экранированные витые пары (STP) типа 1	До 100
100BaseT4	Четыре пары UTP категории 3 (лучше категории 5)	До 100
100BaseFX	Два светодиода оптоволокна с диаметром 62,5/125 мкм	До 400

Увеличение частоты в 10 раз (с 10 Мбит/с до 100 Мбит/с) приводит к тому, что максимальное расстояние между двумя точками сегмента уменьшается в 10 раз, т. е. до 250 м для витой пары (если быть точнее, до 205 м), следовательно в сегмент могут входить максимально два концентратора (рис. 10.6). Поэтому следует помнить, что переход на сеть 100BaseT нельзя рассматривать как механическую замену оборудования.

На рис. 10.6 показаны расстояния для витой пары. При этом возможны разные комбинации расстояний: 100+5+100, 5+100+100, 70+65+70 и т. д. Здесь только важно, чтобы максимальная длина между двумя наиболее удаленными точками сети не превышала бы 205 м для витой пары, и 233 м – для оптоволокна при стандартном подключении. Каждый из концентраторов (см. рис. 10.6)

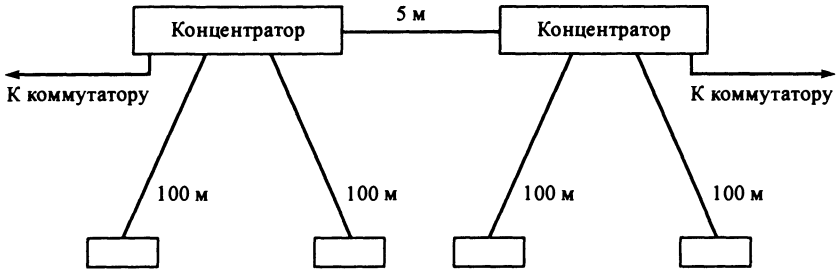


Рис. 10.6. Структура сегмента 100BaseT

можно нарастить, подключив другие концентраторы в стек (до 255 портов на один стек). Если в дополнение к концентраторам 100BaseT использовать и коммутаторы, то сеть можно расширить далеко за пределы одного коммутируемого сегмента.

Сегмент сети 100VG-AnyLan (для краткости будем называть его 100VG) можно построить только по топологии типа «звезда» (рис. 10.7).

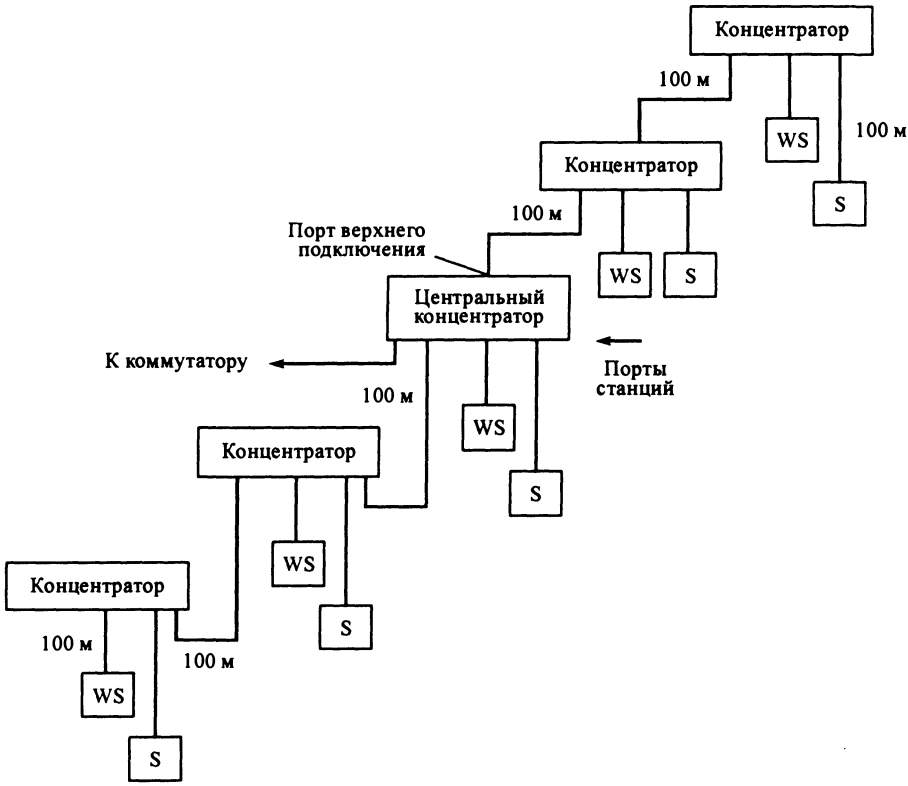


Рис. 10.7. Структура сегмента 100VG-AnyLan

В 100VG используются следующие типы кабелей:

1) четыре неэкранированные витые пары (UTP) категории 3 (3...100 м);

2) две пары UTP категории 5 (5...150 м);

3) две экранированные витые пары (STP) (до 100 м);

4) оптоволокно (до 1 км).

На рис. 10.7 показаны расстояния для неэкранированной витой пары категории 3.

Если узел желает передать какой-либо кадр, он формирует и выставляет запрос к центральному концентратору (ЦК). ЦК периодически осуществляет циклическую проверку своих портов. Если к опрашиваемому порту подключен концентратор нижнего уровня, опрашиваются все порты этого концентратора и т. д. Если нет более приоритетных запросов и сегмент свободен, то узел, выставивший запрос, начинает передачу данных. При этом кадр передается, в отличие от концентратора 100BaseT, только в порт назначения. Одновременно передаются данные только между одной парой узлов. Потери времени на периодический опрос и проведение тестирования соединения составляет примерно 4 % от полного времени работы. Необходимо чтобы каждый концентратор сети должен быть настроен на работу с кадрами одного типа: Ethernet или Token Ring.

При переходе от 10-мегабитных сетей к 100-мегабитным необходимо учитывать особенности этих сетей.

1. Существуют ограничения на топологию сегмента, особенно для 100BaseT (см. рис. 10.6).

2. Проводку кабеля должен осуществлять специалист с использованием специального измерительного оборудования.

3. 100-мегабитные сети дороже 10-мегабитных в несколько раз.

Таблица 10.6. **Преимущества и недостатки Fast Ethernet**

Преимущества	Недостатки
1. Рынок 100BaseT расширяется, и цены на эти сети снижаются быстрее, чем цены на сети 100VG	1. При высокой нагрузке на сегмент Fast Ethernet испытывает большое число коллизий.
2. Используется хорошо зарекомендовавший себя метод доступа CSMA/CD	2. Число концентраторов в сегменте – не более 2
3. Хорошая совместимость оборудования для 100BaseT, выпускаемого разными производителями	3. При использовании витой пары категорий 3 и 5 максимальное расстояние между двумя наиболее удаленными узлами сегмента не должно превышать 205 м (без использования коммутаторов)
4. Более 60 производителей объявили официально о поддержке 100BaseT, объединившись в Альянс быстрого Ethernet (Fast Ethernet Alliance). Среди них такие, как Intel, 3Com, Cabletron Systems, Bay Networks, Cisco, Sun Microsystems, Digital и др.	

Сравнение Fast Ethernet и 100VG-AnyLan

Вопрос выбора 100-мегабитной сети не является тривиальным. В табл. 10.6 и 10.7 перечислены преимущества и недостатки сетей Fast Ethernet и 100VG-AnyLan.

Таблица 10.7. **Преимущества и недостатки 100VG-AnyLan**

Преимущества	Недостатки
1. Число концентраторов в сегменте может достигать 5 2. Максимальное расстояние между двумя наиболее удаленными узлами сегмента при использовании витой пары категории 3 равно 600 м, для витой пары категории 5...900 м (без использования коммутаторов) 3. При переходе от 10-мегабитных сетей (10BaseT) к 100VG-AnyLan достаточно заменить концентраторы и сетевые адаптеры, не меняя топологию сети (в 100VG в каскад можно соединить до 5 концентраторов, в 10BaseT – до 4-х) 4. В сети поддерживается передача кадров Ethernet и Token Ring (но на разных концентраторах) 5. Используется не стандартный «манчестерский» способ кодирования, а кодирование «квартетом», который позволяет передавать два бита за один такт 6. Имеется возможность приоритетной передачи	1. Использование отрезка кабеля категории 5 больше 100 м противоречит стандартам TIA/EIA-568A прокладки кабельных систем 2. Рынок 100VG постоянно сокращается, цены снижаются медленно 3. Только Hewlett-Packard серьезно занимается пропагандой и внедрением технологии 100VG 4. Практически нет приложений, поддерживающих приоритетную обработку

10.3. Выбор магистрали для объединения локальных сетей в черте города

В настоящее время в качестве MAN-сети используют следующие сети:

- FDDI (Fiber Distributed Data Interface) – оптоволоконный интерфейс распределенных данных;
- ATM (Asynchronous Transfer Mode) – режим асинхронной передачи.

Рассмотрим каждый из этих способов построения MAN-сети.

Сети FDDI

Адаптеры FDDI поддерживают метод доступа к сети, основу которого составляет Token Ring. Метод доступа, используемый в сетях FDDI, имеет следующие отличия от Token Ring.

1. В Token Ring маркер передается следующей станции только после возвращения кадра в узел, который передал кадр в сеть. В методе FDDI маркер будет передан непосредственно после передачи кадра данных в сеть.

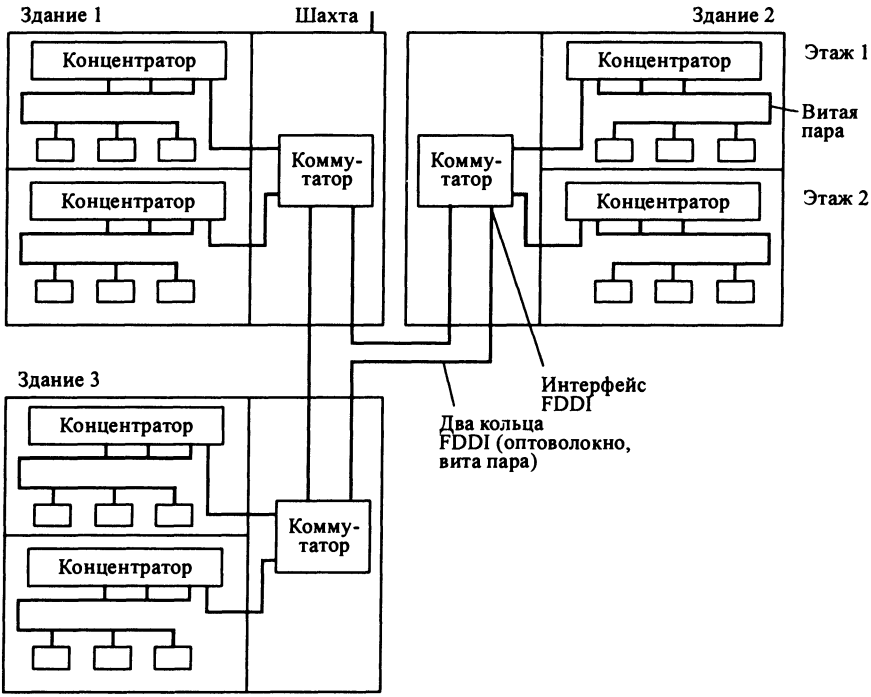


Рис. 10.8. Использование интерфейса FDDI для объединения сетей Ethernet

2. В методе FDDI не используется поле приоритета.

Максимальная длина кольца FDDI с одномодовым оптоволоконным кабелем – 100 км, с многомодовым оптоволоконным кабелем – 20 км (выпускается оборудование FDDI и для витой пары). Скорость передачи по кольцу равна 100 Мбит/с (для дуплексного режима – 200 Мбит/с). К двум кольцам можно подключить до 500 станций или до 100 коммутаторов. Сети FDDI часто используют для объединения нескольких сетей Ethernet, расположенных в разных зданиях (рис. 10.8).

К кольцу можно напрямую подключать и станции, например магистральные высокоскоростные серверы.

Сети ATM

Целью создания сетей ATM явилась попытка предоставления клиентам самых разнообразных услуг. ATM позволяет предоставлять услуги мультимедиа (проведение компьютерных видеоконференций, передача в реальном времени смешанных документов, т. е. содержащих данные, тексты, звук и видеоизображения), передавать видеоинформацию, сигналы телевидения, речь, объединять ЛВС и телекоммуникационные сети.

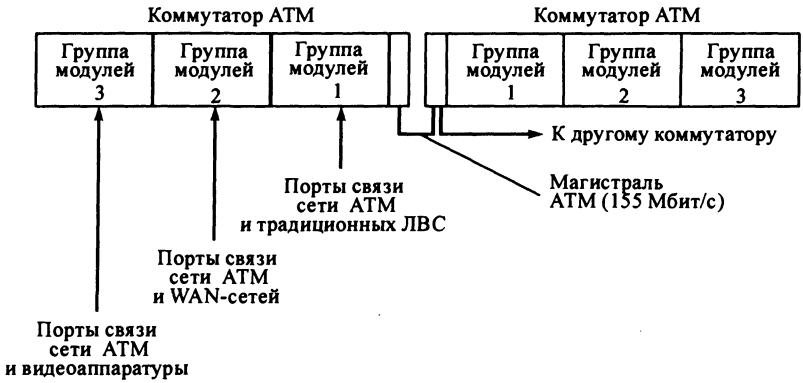


Рис. 10.9. Сеть на базе коммутаторов ATM

Рекомендация ITU-T I.211 делит все услуги ATM на интерактивные (почтовые, диалоговые, «по запросу») и вещательные (телевизионные). Это позволяет сформулировать требования к оконечной аппаратуре и устройствам телекоммуникаций: обеспечение связи «точка-точка», передача вещательных сообщений, предоставление различных классов услуг (ABR, CBR и др.), возможность активного управления со стороны пользователей и др.

Сети ATM строятся на базе коммутаторов ATM (рис. 10.9).

Коммутаторы ATM состоят из модулей. Состав модулей, которые входят в состав коммутатора, определяется фирмой-изготовителем. Для ATM разработаны три основные группы модулей (см. рис. 10.9). Характеристики этих модулей (устройств) приведены в табл. 10.8 – 10.10.

Таблица 10.8. **Характеристики модулей для связи сети ATM и традиционных ЛВС**

Характеристика	Значение	Примечание
Число портов ATM /их тип	1–2/STM–1 (OC–3),155 Мбит/с	Второй порт ATM, как правило, является резервным
Число портов ЛВС /их тип	12–82/Ethernet, Token Ring, FDDI	Порты могут поддерживать несколько адресов MAC (сегментные порты) или только один адрес MAC (для подключения одной станции); 90 % устройств имеют от 12 до 24 сегментных портов Ethernet
Функции матобеспечения	Стандарты UNI 3.0/3.1, LAN Emulation 1.0	В 90 % случаев устройства выполняют только функции клиентов LAN Emulation и требуют для работы внешнего сервера LAN Emulation
Дополнительные функции	LAN Emulation 1.0 (Server), порты ATM 25 Мбит/с	Иногда порты Ethernet являются переключаемыми и могут работать в режиме ATM 25 Мбит/с

Окончание табл. 10.8

Характеристика	Значение	Примечание
Технология коммутации	Коммутатор, быстрый мост, маршрутизатор	Подавляющее большинство устройств выполнены по технологии коммутаторов ЛВС с подключением порта ATM в режиме глобального (up-link)
Наращиваемость	Модульные или стековые устройства	Многие стековые устройства имеют недостаточную пропускную способность внешней шины
Примерная минимальная/ максимальная стоимость на порт Ethernet (в «московских» ценах)	300/1900 долл.	Максимальная стоимость соответствует модульным устройствам с возможностями маршрутизации транспортных протоколов и полным резервированием функций

Таблица 10.9. **Характеристики модулей для связи сети ATM и глобальных сетей (WAN-сетей)**

Характеристика	Значение	Примечание
Число портов ATM /их тип	1 – 4/от E1 (2,048 Мбит/с) до STM–4 (622,08 Мбит/с)	Обеспечен широкий спектр оптических и электрических интерфейсов для большинства применяемых в этом диапазоне скоростей стыков (E3, T3, HSSI)
Число «обычных» портов/их тип	2 – 64/ от 19,2 кбит/с до 52 Мбит/с	Подключение к сетям Frame Relay, X.25; организация выделенных каналов с интерфейсами V.35, E1, E3, HSSI
Функции матобеспечения	Frame Relay, X.25; эмуляция выделенной линии; AAL 1,2,3/4,5; PVC/SPVC; CBR; VBR	К сожалению, на сегодняшний день протоколы взаимодействия АТС плохо поддержаны и для их соединения сеть ATM может предложить только синхронный канал связи без возможности коммутации отдельных телефонных разговоров
Дополнительные функции	Передача видеоинформации; проведение видеоконференции; объединение ЛВС через встроенные мосты	При передаче видеоинформации используется кодирование по протоколам JPEG, MPEG, MPEG-2
Применяемая технология	Раздельные матрицы коммутации, интерфейсные и протокольные модули, объединенные скоростными шинами	Часто применяется шина VME и раздельные матрицы коммутации для уровней адаптации AAL1 и AAL3/4,5

Окончание табл. 10.9

Характеристика	Значение	Примечание
Наращиваемость	Модульные шасси с 5 –20 посадочными местами	Полное резервирование питания и других функций
Примерная минимальная/ максимальная стоимость на порт E1 (в «московских» ценах)	2400/20 000 долл.	Минимальная стоимость соответствует немодульному устройству на 3 порта E1, максимальная – магистральному устройству доступа с полным резервированием функций и возможностью установки более 60 портов E1

Таблица 10.10. **Характеристики модулей для связи сети ATM и видеоаппаратуры**

Характеристика	Значение	Примечание
Число портов ATM/их тип	1/от E1 до STM–1	Обеспечен широкий спектр оптических и электрических интерфейсов (E1, ATM 25 Мбит/с, E3, STM–1)
Число видеовходов/выходов	1 – 6/1 – 4	Видеокодеки имеют больше портов, чем видеодекодеры
Протоколы сжатия	JPEG, MPEG, MPEG-2	Осуществляется сжатие стандартного видеосигнала (VHS, 25 кадр/с) до битового потока со скоростью 2 Мбит/с
Качества сигнала	VHS, SVHS	При передаче сигнала SVHS необходимая полоса пропускания удваивается
Наращиваемость	Автономные или стековые устройства	Стековые устройства, как правило, имеют по одному интерфейсу на блок
Примерная минимальная/ максимальная стоимость на видеопорт (в «московских» ценах)	1100/5000 долл.	Стоимость зависит от применяемых технологий сжатия и набора дополнительных функций

Преимущества и недостатки сетей FDDI и ATM

В настоящее время высокоскоростные магистрали (100 Мбит/с) строят только на основе FDDI и ATM. Все другие широко известные сети (например, 100BaseT) работают на слишком незначительных расстояниях, чтобы их можно было использовать в качестве корпоративной магистрали.

В долгосрочной перспективе ATM позволяет строить корпоративные магистрали большой протяженности, громадной пропускной способности и с невиданными доселе характеристиками. Но сети ATM очень дорогостоящие.

Известно, что протокол FDDI плохо подходит для передачи мультимедиа из-за большого размера кадра FDDI. ATM же неэффективно работает при пе-

ресылке обычных файлов. Во-первых, это связано со значительными потерями Ethernet – ATM. Во-вторых, 5 байт из 53 в кадре ATM занято под системную информацию (заголовок). Таким образом, если в FDDI накладные расходы на канальном уровне составляют порядка 0,5 %, то у ATM – 10 %.

Для магистрали чаще всего предлагаются коммутаторы ATM с пропускной способностью 155 и 622 Мбит/с. Но из-за больших накладных расходов ATM с пропускной способностью 155 Мбит/с уступают в производительности FDDI для обычных сетевых потоков. Корпоративная магистраль на 622 Мбит/с кажется неплохим вариантом, если бы не высокая цена.

Некоторые сетевые интеграторы рекомендуют подключать все серверы рабочих групп к магистрали FDDI напрямую, а клиентов – через маршрутизатор (или коммутатор с функциями маршрутизатора). Такой выбор они аргументируют тем, что в этом случае клиент сети имеет доступ к любому серверу предприятия максимум через одно маршрутизирующее устройство. В качестве контраргумента можно привести наблюдение специалистов фирмы Bay Networks: в хорошо спроектированной сети только 20% трафика приходится на межсетевое взаимодействие. Иными словами, основной трафик внутри рабочей группы связан с серверами этого же подразделения. Незачем забивать магистраль сетевыми потоками, не относящимися к корпоративным приложениям. Таким образом, серверы рабочих групп лучше подключать к магистрали через коммутатор (маршрутизатор) (см. рис.10.8). Серверы, обслуживающие все предприятие, необходимо подключать к магистрали напрямую.

10.4. Выбор магистрали WAN для объединения сетей в разных городах

В настоящее время в России для построения WAN-сетей (глобальных сетей) в основном используют сети X.25, Frame relay и ISDN.

Сети X.25

Сетями X.25 называют сети пакетной коммутации, доступ к которым выполняется в соответствии с рекомендациями МККТТ X.25. На сегодняшний день, несмотря на появление новых интегрированных технологий сетей передачи данных и сетей связи, рассчитанных на высокоскоростные каналы связи, сети X.25 по-прежнему наиболее распространены. На базе протокола X.25 работают такие отечественные сети, как ГЛАСНЕТ, ИАСНЕТ, РОСПАК, СПРИНТ и др. На рис. 10.10 представлена структура сети X.25.

Для подключения локальной сети к сети X.25 можно использовать следующие варианты подключения.

1. В сервер или PC встраивается специальная плата и устанавливается специальное программное обеспечение.

2. К сети подключается специальное автономное устройство (мост/ маршрутизатор) удаленного доступа, поддерживающее протокол X.25.

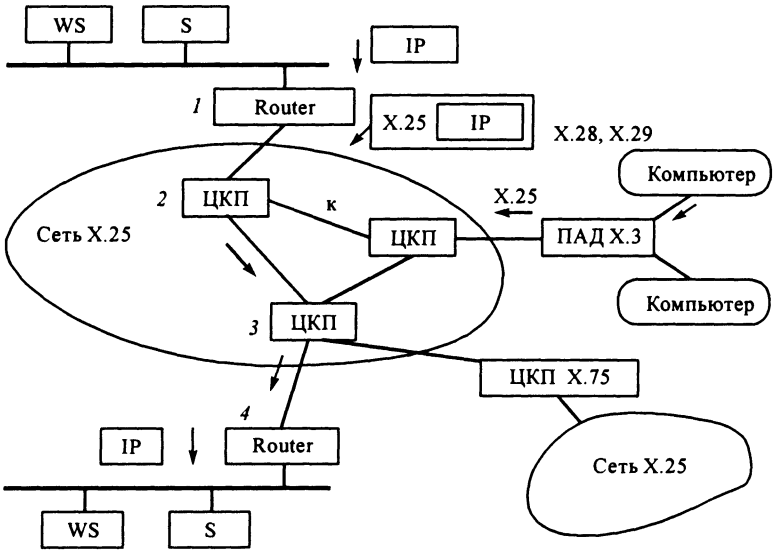


Рис. 10.10. Структура сети X.25

Преимущество автономных устройств над встраиваемыми в компьютер платами, помимо большей производительности, заключается в том, что они не требуют установки специального программного обеспечения и сопрягаются с локальной сетью по стандартному интерфейсу, что позволяет реализовать более гибкие и универсальные решения.

Если требуется подключить компьютер к сети X.25 в монопольном режиме (без сети), то подключение осуществляется по стандартам X.3, X.28, X.29. Эти стандарты определяют функционирование специальных устройств доступа к сети – сборщиков/разборщиков пакетов (PAD – Packet Assembler/ Disassembler). На практике используется термин ПАД (см. рис. 10.10). ПАД используют для доступа к сети абонентов при асинхронном режиме обмена информацией, т. е. , например, через последовательный порт компьютера (непосредственно или с применением модемов). ПАД обычно имеет несколько асинхронных портов и один синхронный (порт X.25). Он накапливает поступающие через асинхронные порты данные, упаковывает их в пакеты и передает их через порт X.25. Конфигурируемые параметры ПАД определяются выполняемыми задачами. Эти параметры описываются стандартом X.3. Совокупность параметров носит название «профайла» (profile). Стандартный набор состоит из 22 параметров, функциональное назначение этих параметров одинаково для всех ПАД. В профайл входят параметры, задающие скорость обмена по асинхронному порту, параметры, характерные для текстовых редакторов (символ удаления знака и строки, символ вывода на экран предыдущей строки и т. п.), параметры, включающие режим автоматической добавки строки незначащи-

ми символами (для синхронизации с медленными терминалами), а также параметр, определяющий условие, при выполнении которого формирование пакета заканчивается.

Параметры, описывающие канал X.25, являются немаловажными и для узловых элементов собственно сети X.25, называемыми центрами коммутации пакетов (ЦКП). Однако ими список параметров ЦКП не исчерпывается. В процессе конфигурации ЦКП необходимо заполнить таблицу маршрутизации (routing table), позволяющую определить, на какой из портов ЦКП направляются поступившие в них пакеты в зависимости от адресов, содержащихся в этих пакетах. В таблице задаются как основные, так и альтернативные маршруты. Разные образцы оборудования ЦКП отличаются алгоритмами перехода к альтернативному маршруту, а также допустимым количеством таких маршрутов. В некоторых типах оборудования переход к альтернативному маршруту происходит только в случае отказа одного из звеньев основного маршрута, т. е. при надежной работе сети маршрут (виртуальный канал) передачи данных между двумя оконечными узлами не изменяется. На рис. 10.10 маршрут *1–2–3–4* является примером виртуального канала между маршрутизаторами *1* и *4*. Для другого оборудования переход от одного маршрута к другому происходит динамически в зависимости от загруженности маршрутов. При этом решение принимается на основании многопараметрической формулы (оборудование фирмы Motorola ISG).

Важной функцией некоторых ЦКП является функция стыковки сетей (шлюза между сетями). Действительно, в мире существует множество сетей X.25 общего пользования, частных, ведомственных. Естественно, в различных сетях могут быть установлены разные значения параметров передачи по каналам X.25 (длина кадра и пакета, система адресования и др.). Для того, чтобы все эти сети могли стыковаться друг с другом, была разработана рекомендация X.75, определяющая правила согласования параметров при переходе из сети в сеть. Сопряжение соседних сетей рекомендуется производить через ЦКП, в котором с достаточной полнотой реализована поддержка шлюзовых функций. Такой ЦКП должен, например, транслировать адреса при переходе пакета из одной сети в другую. Эта функция обычно реализуется с помощью конфигурации специальной таблицы трансляции адресов в шлюзовом ЦКП. Для ЦКП, не сопрягающихся с узлами другой сети, наличие шлюзовых функций не является обязательным.

Обычно коммерческие сети создаются по следующей схеме:

- закупка ЦКП,
- аренда помещения в местных АТС, где устанавливаются ЦКП,
- аренда телефонных каналов связи (аналоговые или цифровые) у государственных или других коммерческих компаний,
- настройка оборудования.

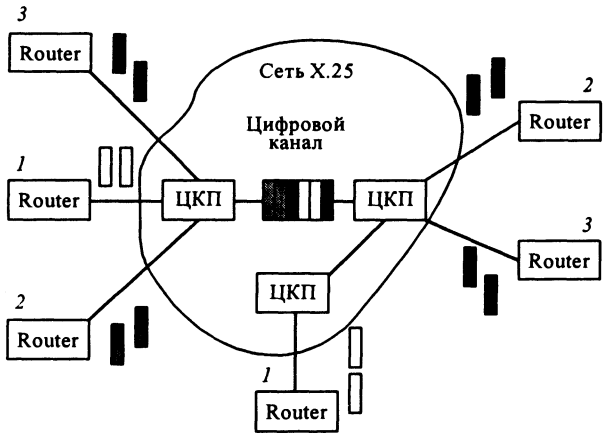


Рис. 10.11. Совместное использование канала связи

Отметим некоторые важные достоинства сетей X.25.

1. Сети X.25 обеспечивают раздельное использование дорогих цифровых каналов связи (рис. 10.11). Здесь виртуальные каналы *1–1*, *2–2*, *3–3* совместно используют (разделяют) цифровой канал связи.

2. В качестве каналов связи, соединяющих соседние ЦКП (на рис. 10.10 в качестве примера такой канал обозначен буквой «к»), могут выступать выделенные (аналоговые или цифровые) и коммутируемые телефонные линии связи.

3. Наличие альтернативных маршрутов обеспечивает высокую надежность передачи данных.

Но при всех достоинствах сетевой технологии, базирующейся на протоколе X.25, у нее есть и свои ограничения.

1. Автономные мосты/маршрутизаторы, сетевые карты, ПАДы оконечных устройств могут передавать данные со скоростью до 64 кбит/с.

2. Для обеспечения высокой скорости передачи требуется выделенная линия от конечного пользователя до ближайшей АТС, где установлено оборудование сети X.25.

3. По сетям X.25 нельзя передавать такие виды информации, как голос и видео (это ограничение преодолевается в технологии, базирующейся на протоколе Frame relay).

Накоплен большой опыт использования сетей X.25. Известно, что применение сетей X.25 эффективно для широкого спектра задач передачи данных. Среди них и обмен сообщениями между пользователями, и обращение большого числа пользователей к удаленной базе данных, а также к удаленному хосту электронной почты, связь локальных сетей, объединение удаленных кассовых автоматов и банкоматов. Иными словами, все приложения, в которых трафик сети не является равномерным по времени.

Сети Frame relay

Технология Frame relay (FR) изложена в § 4.3. Сети Frame relay (ретрансляция кадров) также являются сетями пакетной коммутации, но отличаются от сетей X.25:

на канальном уровне не выполняется контроль ошибок. Контроль за правильностью передачи данных от отправителя должен осуществляться на более высоком уровне иерархии протоколов;

мультиплексирование (маршрутизация) осуществляется на канальном (аппаратном) уровне. Управление потоком отсутствует. В основном применяются постоянные виртуальные каналы.

На рис. 10.12 представлена структура сети Frame relay.

Так как в FR применены виртуальные каналы (статическое мультиплексирование), то абонент (маршрутизатор) имеет возможность в течение некоторого времени передавать данные со скоростью выше той, которая ему гарантируется. В связи с этим главной причиной потери передаваемых данных в

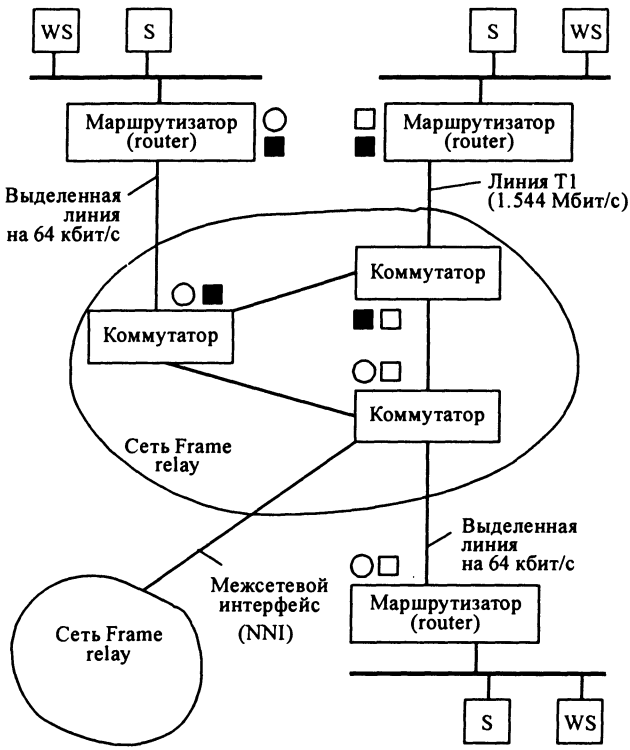


Рис. 10.12. Структура сети Frame relay:
символами ○, ■ и □ обозначены коммутаторы и маршрутизаторы, которые образуют соответствующие постоянные виртуальные каналы

сетях ретрансляции кадров является перегрузка (congestion) узлов коммутации. Управление трафиком организовано так, что абонент по своему выбору ведет передачу либо в гарантированном режиме, либо с превышением заранее согласованной скорости, что, естественно, сопряжено с риском потери информации и с повтором передачи искаженных кадров.

Пропускная способность сети FR, выделяемая виртуальному каналу, характеризуется следующими параметрами.

- гарантированная скорость передачи данных, т. е. обеспечиваемая абоненту постоянно (committed information rate, CIR);
- учетный период – промежуток времени (секунды), для которого определен максимальный объем данных (биты), передаваемых сетью с удовлетворительной вероятностью (committed rate measurement interval, T_c).
- гарантированный объем передачи – максимальный объем данных (биты), транспортировка которых в течение учетного периода T_c обеспечена с высокой вероятностью (committed burst size, B_c).
- дополнительный объем передачи – максимальный объем данных (биты), доставка которых в течение учетного периода T_c (в дополнение к объему B_c) возможна, но с меньшей вероятностью (excess burst size, B_e).
- максимальная скорость передачи данных (excess information rate, EIR), которая определяется как EIR = $(B_c + B_e)/T_c$. Другое название этого параметра – пропускная способность порта (port speed).

Из приведенных определений понятно, что CIR, B_c и T_c должны удовлетворять следующему отношению: CIR = B_c/T_c. Пользователь выбирает (и оплачивает) пропускную способность порта (EIR) и гарантированную скорость передачи данных (CIR) для каждого виртуального канала, проходящего через порт.

Скорость передачи данных вычисляется узлом доступа к сети FR путем измерения объема, переданного за время T_c. При этом выполняются следующие действия:

1. Если полученное значение скорости не превосходит CIR, кадры передаются без изменения.

2. Если скорость больше CIR, но меньше EIR, то в кадрах устанавливается бит DE (Discard Eligibility), разрешающий их удаление (при возникновении перегрузки сети такие кадры отбрасываются в первую очередь). Бит DE может устанавливаться и оборудованием пользователя, которое, таким образом, выбирает, какими кадрами пожертвовать прежде всего.

3. В случае, когда скорость превосходит EIR, поступающие кадры удаляются независимо от каких-либо условий.

Некоторые поставщики услуг предлагают значительные скидки за передачу кадров с битом DE. При наличии в сети достаточного запаса пропускной способности абонент может снизить свои финансовые затраты (иногда больше 50 %), положив CIR = 0 (в этом случае DE = 1 во всех передаваемых кадрах). Таким образом, в сетях FR допускается передача данных со скоростью выше гарантированной вплоть до пропускной способности порта, но при этом некоторые кадры могут быть потеряны и для их восстановления требуется повторная передача.

Российские абоненты могут воспользоваться некоторыми международными службами:

- Global Managed Data Service (английская компания Cable&Wireless PLC),
- SITA (английская компания SITA Group),
- Datanet (финская компания Telecom Finland).

Есть и отечественные сети, предоставляющие услуги Frame relay: Макомнет, Метроком, Роском, СОВАМ-телепорт, Спринт и др.

Диапазон параметра пропускной способности порта EIR составляет от 56 – 64 кбит/с до 1,544 Мбит/с с шагом 64 кбит/с, а CIR – 4, 8, 16, 32, 56, 64 кбит/с и далее до 1,544 Мбит/с с шагом 64 кбит/с.

Основными преимуществами сетей Frame relay являются:

- высокая скорость передачи. В настоящее время сети Frame relay обеспечивают скорость передачи 56 кбит/с и 1,544 Мбит/с;
- малая сетевая задержка при активизации виртуального канала;
- хорошая связность для звездной и ячеистой топологии;
- эффективное использование полосы пропускания.

В то же время можно отметить следующие недостатки Frame relay:

- для подключения к сети Frame relay пользователю необходимо арендовать или иметь собственную выделенную линию;
- для эффективной работы сети требуется высокая надежность каналов связи. Поэтому для построения сетей Frame relay используются дорогие спутниковые, оптоволоконные, цифровые каналы связи;
- сети Frame relay не рассчитаны на передачу больших файлов данных (порядка 100 Мбайт), данных мультимедиа и на обслуживание ровного трафика (например, при коллективной разработке ПО).

Сети Frame relay предназначены, прежде всего, для приложений со случайными сильными всплесками трафика, которые, например, имеют место в сетях электронной почты, автоматизированного проектирования, а также в системах клиент/сервер.

СПИСОК ЛИТЕРАТУРЫ

1. *Блэк Ю.* Сети ЭВМ: протоколы, стандарты, интерфейсы. М.: Мир, 1990.

2. *Олифер В.Г., Олифер Н.А.* Компьютерные сети. Принципы, технологии, протоколы. СПб.: «Питер», 2000.

3. *Новиков Ю.В., Кондратенко С.В.* Локальные сети: архитектура, алгоритмы, проектирование. М.: Эном, 2000.

4. *Мартин Дж.*, Чапмен К.К., Либен Д. Архитектура и реализация АТМ. М.: Лорн, 2000.

5. *Гольдштейн Б.С.* Протоколы сети доступа. Т. 2. М.: Радио и связь, 1999.

6. *Норенков И.П., Трудоношин В.А.* Телекоммуникационные технологии и сети. М.: Изд-во МГТУ им. Н.Э.Баумана, 2000.

7. *Халсалл Ф.* Передача данных, сети компьютеров и взаимосвязь открытых систем. М.: Радио и связь, 1995.

8. Справочник. Протоколы информационно-вычислительных сетей. М.: Радио и связь, 1990.

9. *Джордейн Р.* Справочник программиста персональных компьютеров типа IBM PC XT/AT. М.: Финансы и статистика, 1991.

10. *Фролов А.В., Фролов Г.В.* Программирование модемов. М. «ДИАЛОГ-МИФИ», 1993.(Б-ка системного программиста. Т. 4).

11. *Хаммел Р.* Последовательная передача данных. Руководство для программиста. М.: Мир, СК ПРЕСС 1996.

12. *Элизабет Кларк.* Стандарты и протоколы Интернета //LAN/Журнал сетевых решений. М., 1999. Т. 5. № 2.

13. Microsoft TCP/IP. Изд. «Русская редакция», 1999.

14. *Девис Р.* Руководство по программированию в NetWare/386. М: Изд-во АО «ИСМ», 1994.

15. *Дэй М., Кунц М., Маршалл Д.* Программирование NLM в NetWare 4.0. М: Изд-во «ЛОРИ», 1994.

16. *Нанс Б.* Программирование в локальных сетях. Пермь: Изд-во Перм. ун-та, 1992.

17. *Шабалин А.Р.* Интерфейс пользователя с системой //Технологии электронных коммуникаций. М., 1992. Т. 24.

18. *Казаков С.И.* Основы сетевых технологий. М.: Микроинформ, 1995.

19. *Бычков И.В., Григорьев Ю.А., Левен И.Э.* Операционная система NetWare 4.x. М: Изд-во «МГТУ-ИНТЕРПРОКОМ», 1995.

20. *Пьянзин К.* IntranetWare – Novell держит удар Microsoft// LAN/Журнал сетевых решений. М., 1997. Т. 3. № 1.С. 72 – 77.

21. *Штайнке С.* Построение сетей с серверами NetWare и NT// LAN/Журнал сетевых решений. М., 1998. Т. 4. № 9. С. 99 – 103.

22. *Пьянзин К.* Службы каталогов Novell и Microsoft// LAN/Журнал сетевых решений. М., 1999. Т. 5. № 3. С. 81 – 90.

23. *Пьянзин К.* NetWare 5 – новая ставка Novell// LAN/Журнал сетевых решений. М., 1998. Т. 4. № 12. С. 71 – 79.

24. *Ценк А.* Novell NetWare 4.x. К.: Торгово-издательское бюро BHV, 1996.

25. *Лоренс Б.* Novell NetWare 4.1 в подлиннике. СПб: BHV – Санкт-Петербург, 1996.

26. *Григорьев Ю. А., Фраерман В. В.* NetWare 5. Настольная книга администратора. М.: ДМК, 2000.

27. *Цуканов Ю.П.* Англо-русский словарь аббревиатур в области информационных технологий /Под ред. П.С.Иванова. М.: Изд-во «ИнфоАрт», 1995.

28. ITU-T Recommendation Q.920 (I.440), ISDN User – Network Interface – Data Link Layer – General Aspects. Geneva, 1993.

29. ITU-T Recommendation Q.921 (I.441), ISDN User – Network Interface – Data Link Layer Specification. Geneva, 1993.

30. ITU-T Recommendation X.25, Data Networks and Open System Communication – Public Data Networks – Interfaces. Geneva, 1989.

ЗАКЛЮЧЕНИЕ

В предлагаемом учебном пособии авторы попытались раскрыть сущность основ телекоммуникаций и сетевых технологий информатизации человеческой деятельности на современном этапе. Несмотря на то что значительный объем знаний в этой динамической области техники быстро устаревает, остается еще целый пласт базовых знаний, который составляет фундамент образования любого специалиста в области автоматизированных систем обработки информации и управления.

Независимо от того, какие технологии будут применяться в локальных и глобальных сетях через 5 – 10 лет, данные будут передаваться на основе метода коммутации пакетов, которые могут называться и иначе – кадрами, ячейками или как-нибудь еще, но суть метода от этого не изменится. Коммуникационные протоколы будут образовывать иерархический стек, а надежность передачи данных обеспечивается за счет повторной передачи кадров, ошибки в которых будут обнаруживаться на основе методов физического и логического кодирования.

К сожалению, ограниченный объем книги не позволил включить в нее разделы учебного лабораторного практикума. По этой же причине не вошли материалы по беспроводным сетям, основанным на методе кодового разделения каналов CDMA и другим перспективным, по оценкам специалистов, технологиям, например, недавно появившаяся технология пакетной передачи данных (GPRS) в сетях GSM – это цифровая передача голоса и заполнение пауз в промежутках между разговорами абонентов передачей пакетов данных. Однако для понимания новых технологий достаточно увидеть в них комбинацию базовых идей.

Известно, что знание основополагающих концепций позволяет легко разбираться в новых, пусть даже, на первый взгляд, и очень сложных технических решениях и технологиях. Авторы надеются, что учебное пособие, которое вы прочитали, обогатило запас ваших базовых знаний. Пусть эти знания станут тем инструментом, с помощью которого вы сможете обновлять знания о постоянно изменяющемся мире телекоммуникаций и компьютерных сетей.

СПИСОК ОСНОВНЫХ АНГЛОЯЗЫЧНЫХ СОКРАЩЕНИЙ

AAL – ATM Adaptations Level – уровень адаптации ATM

ABM – Asynchronous Balanse Mode – Асинхронный сбалансированный режим

ACK – ACKnowledgments – подтверждение приема

ADC – Advanced Digital Communication – усовершенствованная цифровая система мобильной радиотелефонной связи (стандарт сотовой связи в США) – D-AMPS или Analog-to-Digital Converter – аналого-цифровой преобразователь

ADM – Asynchronous Disconnection Mode – асинхронный режим разъединения

ADP – Answerer Detection Pattern – «шаблон обнаружения отвечающего» в протоколе LAPM

ADPCM – Adaptive Differential Pulse Code Modulation – стандарт МККТТ кодирования голоса

ADPCM – Adaptive Differencial PCM – адаптивная ДИКМ, оцифровывается не сам сигнал, а его отклонение от предсказанного значения

AM – Amplitude Modulation – амплитудная модуляция

AMI – Bipolar Alternate Mark Inversion – метод биполярного кодирования с альтернативной инверсией

AMPS – Advanced Mobile Phone System – усовершенствованная система мобильной радиотелефонной связи (стандарт сотовой связи в США)

ANSI – American National Standards Institute – Американский национальный институт стандартов

API – Application Programs Interface – интерфейс прикладного программирования, набор функций, доступных прикладным программам и обеспечивающих построение пользовательского интерфейса, обмен информацией, внутрисистемный сервис и т. д.

APPC – Advanced Program-to-Program Communication – одноранговый SNA-протокол фирмы IBM

ARM – Asynchronous Response Mode – режим асинхронного ответа

ARP – Address Resolution Protocol – протокол разрешения адресов

ARQ – Automatic Repeat reQuest – метод автоматического повтора запроса

ASMP – ASymmetric MultiProcessing – ассиметричная многопроцессорная обработка

ATM – Asynchronous Transfer Mode – режим асинхронной пересылки (передачи)

AU – Access Unit – устройство доступа

AUI – Access Unit Interface – интерфейс с устройством доступа

BA – Balanced Asynchronous – сбалансированная конфигурация

BER – Bit Error Rate – интенсивность битовых ошибок

BNC – British Naval Connector – соединитель для тонкого коаксиального кабеля

BRI – Basic Rate Interface – интерфейс базового уровня в сетях ISDN

BSC – Binary Synchronous Communications – двоичная синхронная передача данных или Binary Synchronous Control – двоичное синхронное управление, протокол двоичной синхронной передачи данных фирмы IBM

BSS – Base Station System – оборудование базовой станции сотовой связи подвижной

BT – bit time – битовый интервал (в Fast Ethernet 1 BT = 0,01 мкс)

BTS – Base Telecommunication Station – базовая станция в сети сотовой подвижной связи

BUS – Broadcast and Unknown Server – сервер широковещательной рассылки и идентификации неопознанных ресурсов в LANE

CCITT – Consultative Committee on International Telephony & Telegraphy – международный консультативный комитет по телеграфии и телефонии (МККТТ)

CDPD – Cellular Digital Packet Data – пакеты цифровых данных сотовой сети – стандарт на передачу пакетов, использующий существующую инфраструктуру аналоговой сотовой телефонной связи

CD-ROM – Compact Disk Read Only Memory – ПЗУ на компакт-дисках

CIR – Committed Information Rate – согласованная информационная скорость, параметр передается в пакете запроса на установление соединения в сетях Frame relay

CLP – cell loss priority – приоритет потери ячейки, параметр ячейки в сетях ATM

CPN – Customer Premises Node – узел на территории потребителя в сетях ATM

CRC – Cyclic Redundancy Check – циклический избыточный контроль

CSMA/CD – Carrier Sense Multiple Access / Collision Detection – множественный доступ с контролем носителя и обнаружением столкновений, МДКН/ОС или Communication(s) Servise Unit – связное устройство

CSU – Channel Service Unit – устройство обслуживания канала или Communication(s) Servise Unit – связное устройство

DA – Destination Address – адрес получателя

D-AMPS – Digital AMPS – цифровая усовершенствованная система мобильной радиотелефонной связи (стандарт сотовой связи в США)

DCE – Data Circuit terminating Equipment – оконечное оборудование канала передачи данных (аппаратура, обеспечивающая кодирование и преобразование сигналов между DTE и линией связи, например, модем или Data Communications Equipment – аппаратура передачи данных, АПД или Distributed Computing Environment – среда распределенных вычислений

DET – Directory Entry Table – таблица записей каталога

DID – Directory Information Base – информационная база объектов

DIT – Directory Information Tree – дерево информационного каталога

DL – Distribution Lists – списки рассылки

DLCI – Data Link Connection Identifier – номер виртуального соединения в технологии Frame relay

DLE – Data Link Escape – символ байтстаффинга в байт-ориентированных протоколах

DNIC – data network identification code – код идентификации сети

DPCM – Differencial PCM – дифференциальная (разностная) ИКМ (ДИКМ), когда вместо кодирования отсчетов кодируются разности между соседними отсчетами

DPP – Distributed Paralltl Processing – распоределенная параллельная обработка

DPSK – Differential Phase Shift Keying – фазоразностная модуляция

DSA – Directori System Agent – системный агент доступа к каталогу

DSAP – Destination Service Access Point – точка доступа к сервису получателя, определяет тип протокола сетевого уровня станции-получателя в сетях Ethernet

DSB – double sidebaud – двухполосная система, т. е. система с АМ, которая передает обе боковых и несущую частоту

DSU – Data Service Unit – устройство обработки данных

DTE – Data Terminal Equipment – оконечное (терминальное) оборудование обработки данных

DTI – Digital Trunk Interface – интерфейс цифрового магистрального канала

DUA – Directory User Agent – пользовательский агент доступа к каталогу

EIA – Electrical Industry Association – Ассоциация электрической промышленности США

EIR – Equipment Identification Register – регистр идентификации оборудования

ELAN – Emulation LAN – локальная сеть, эмулируемая ATM-сетью по спецификации LANE

EOT – End Of Transmission – символ конца передачи

ES – End Systems – оконечные системы

ETC – Enhanced Throughput Cellular – усовершенствованная сотовая связь – протокол фирмы AT&T, обеспечивающий исправление ошибок в сотовых сетях

ETX – End of TeXt – символ конца текста, обычно определяет конец кадра в байт-ориентированных протоколах, в коде ASCII символ ETX – 03h имеет двоичное значение 00000011

FAT – File Allocation Table – таблица размещения файлов

FC – Fibre Channel – оптический канал

FCS – **Frame Check Sequence** – контрольная сумма всех полей кадра Ethernet (за исключением полей преамбулы, признака начала кадра и самой контрольной суммы)

FDDI – Fiber Distributed Data Interface – распределенный интерфейс передачи данных по волоконно-оптическим каналам (стандарт ANSI, модифицированный IEEE/ISO

FDM – Frequency Division Method – метод частотного разделения

FM – Frequency Modulation – частотная модуляция

FOIRL – Fiber Optic Inter-Repeater Link – волоконно-оптическая связь между ретрансляторами – протокол передачи данных по ВОЛС в сетях Ethernet

FRAD – Frame Relay Access Device – устройства доступа к сети Frame relay

FS – File Server – файловый сервер

FSK – Frequency Shift Keying – метод частотной модуляции, в котором несущая переключается сигналами с одной частоты на другую при неизменной амплитуде

FTAM – File Transfer Access and Management – протокол OSI доступа к файлам

FTP – File Transfer Protocol – протокол для передачи файлов

GFC – Generic Flow Control – Управление потоком (поле ячейки ATM), используется только при взаимодействии конечного узла и первого коммутатора сети ATM

GFI – general format identifier – идентификатора универсального формата в заголовке 3-го уровня сети X.25

GMII – Gigabit Media Independent Interface – независимый от среды интерфейс физического уровня сетей Gigabit Ethernet

GSM – Global System for Mobile communications – глобальная (общеевропейская) сотовая система цифровой радиосвязи (стандарт)

HCSS – High Capacity Storage System – система накопителей высокой емкости

HDLC – High-level Data Link Control – управление каналом передачи данных высокого уровня (протокол)

HDSL – High-bit-rate Digital Subscriber Line – высокоскоростная цифровая абонентская линия, технология, обеспечивающая передачу на скорости 1,536 или 2,048 Мбит/с в обоих направлениях по четырехпроводной абонентской линии до 3,7 км

HEC – header error check – контроль ошибок заголовка

HST – High Speed Technology – технология высокой скорости, реализующая асимметричный дуплексный модемный протокол с частотным разделением каналов

ICMP – Internet Control Message Protocol – протокол межсетевых управляющих сообщений

IDC – Insulation Displacement Connector – соединитель со сдвигом изоляции

IDE – Integrated Developer(s) Enviroment – интегрированная среда разработчика, предложена фирмой Borland, или Integrated Drive (Disk) Electronics – встроенная электроника управления диском (стандарт), или Integrated Drive Eguipment – интеллектуальное оборудование дискового накопителя, популярный интефейс, поддерживающий до двух жестких дисков

IEEE – Institute of Electrical and Electronics Engineers – Институт инженеров по электротехнике и радиоэлектронике США

IMSI – International Mobile Station Identifier – международный идентификационный номер

IP – Internet Protocol – межсетевой протокол

IPG – Inter Packet Gap – промежуток времени между окончанием одного пакета и началом следующего

IPM – Inter Personal Message – интерперсональное сообщение, состоит из заголовка и тела, форматы IPM различаются набором поддерживаемых типов данных и правил кодирования текста, содержащего символы национальных алфавитов, используется в системах X.400

IPN – Inter Personal Notification – интерперсональная нотификация, используется для автоматического уведомления отправителя о факте доставки и/или прочтения, посланного им сообщения в системах X.400

IPX – Internetwork Packet eXchange – межсетевой обмен пакетами, простейший стандартный протокол сети NetWare, выполняющий функции транспортного и сетевого уровня без управления потоком и обнаружения ошибок; разработан фирмой Novell

IRF – Inherited Right Filter – фильтр наследуемых прав

IRQ – InterRuption reQuest – запрос на прерывание

IS – Initialization Status – состояние инициализации

ISDN – Integrated Services Digital Network – цифровая сеть с комплексными услугами

ISO – International Standard Organization – Международная организация по стандартам

ISU – Information Symbol Unit – единица передачи информации в сетях ARCNet

ITS – Information Transfer Status – состояние передачи информации

ITU-T – International Telecommunications Union – Technical standards sector, Международный телекоммуникационный союз – Сектор технических стандартов, международная организация, создающая стандарты для телекоммуникаций

JDC – Japan Digital Communication – цифровая система мобильной радиотелефонной связи (стандарт сотовой связи в Японии)

JPEG – Joint Photographic Experts Group – объединенная экспертная группа по фотографии, или алгоритм сжатия неподвижного изображения, разработанный этой группой, или стандарт JPEG и соответствующий формат файлов

LAN – Lokal Area Network – локальная (вычислительная) сеть, ЛВС

LANE – LAN Emulation – эмуляция локальных сетей, спецификация, разработана ATM Forum для технологии ATM

LAPB – Link Access Procedure, Balanced – процедура доступа к каналу связи, сбалансированная

LAP-F – процедура доступа к каналу связи Frame relay

LAPM – Link Access Procedure for Modems – процедура доступа к каналу связи для модемов

LAPD – Link Access Procedure, D channel – процедура доступа к каналу связи, D-канал

LCI – logical channel identifier – идентификатора логического канала в заголовке 3-го уровня сети X.25

LDAP – Lightweight Directory Access Protocol – облегченный протокол доступа к каталогу

LDS – Logical Disconnection Status – состояние логического разъединения

LE_ARP – LAN Emulation Address Resolution Protocol – протокол разрешения адресов в спецификации LANE

LEC – LAN Emulation Client – эмуляция LAN клиентская часть, программный компонент LANE

LECS – LAN Emulation Configuration Serve – сервер конфигурации в LANE

LES – LAN Emulation Server – эмуляция LAN серверная часть, программный компонент LANE

LLC – Logical Link Control – управление логическим каналом

LPC – Linear Predictive Coding – кодирование голоса методом прогнозирования направления изменения амплитуды сигнала

LUNI – LAN Emulation User-Network Interface – интерфейс пользователь – сеть в LANE, протокол взаимодействия клиентской части протокола LANE, а именно LEC, с серверными частями этой спецификации LECS, LES и BUS

LZ – Lempel-Ziv – алгоритм сжатия без потерь Лемпеля-Зива

MAC – Medium Access Control – управление доступом к передающей среде (УДС)

MAN – Metropolitan Area Networks – городские сети, сети метрополий

MAU – Media Attachment Unit – блок доступа к среде в сетях Token Ring

MHE – Messaging Handling Environment – среда управления сообщениями

MHS – Messaging Handling Serviice – служба обработки сообщений, протокол фирмы Novell для связи с системами электронной почты или Messaging Handling System – система обработки сообщений

MII – Media Independent Interface – независимый от среды интерфейс физического уровня сетей Fast Ethernet

MIME – Multipurpose Internet Mail Extensions – многофункциональные расширения почты Internet

MNP – Microcom Networking Protocol – сетевые протоколы фирмы Microcom

MPC – MPOA Client – клиент MPOA

MPEG – Motion Picture Experts Group – экспертная группа по кинематографии, или одноименный алгоритм сжатия подвижного изображения, а также соответствующий формат файлов

MPOA – Multi Protocol Over ATM – мультипротокол поверх ATM, система передачи пакетов сетевого уровня через сеть ATM в режиме LANE

MPS – MPOA Server – сервер MPOA

MS – Mobile Station – подвижная станция или Message Stope – хранилище сообщений

MSAU – Multi-Station Access Unit – блок множественного (многостанционного) доступа к среде в сетях Token Ring

MSC – Mobile Switching Centre – центр коммутации подвижной связи

MSL – Mirrored Server Link – задублированная связь серверов

MTA – Message Transfer Agent – агент передачи сообщений

MTS – Message Transfer System – система передачи сообщений

MTU – Maximum transfer Unit – максимальная единица передачи данных, определяет максимальное значение, которое может иметь длина поля данных протокола

NAC – Negative ACKnowledgments – неготовность к приему

NCP – NetWare Core Protokol – протокол транспортного уровня фирмы Novell, является надстройкой над протоколом сетевого уровня IPX

NDM – Normal Disconnection Mode – нормальный режим разъединения

NDS – NetWare Directory Services – (глобальная) служба каталогов ОС NetWare, единый сетевой каталог, набор функций, предоставляющий пользователям «сквозной» доступ к распределенным сетевым ресурсам; имеет древовидную структуру, основан на стандарте X.500; позволяет объединить все сетевые ресурсы в единую систему независимо от их физического размещения

NEMA – National Electrical Manufacturers Assotiation – Национальная ассоциация производителей электротехнической промышленности

NEXT – Near End Cross Talk – перекрестные наводки на ближнем конце

NHRP – Next Hop Resolution Protocol – протокол определения пути между серверами через сеть ATM

NLM – NetWare Loadable Module – загружаемый модуль ОС NetWare (приложения, выполняемые на сервере локальной сети под управлением сетевой ОС NetWare) фирмы Novell

NLSP – NetWare Link Services Protokol – протокол обслуживания связей (маршрутизации) в ОС NetWare

NMS – NetWare Management System – система управления сетями ОС NetWare

NNI – Node Network Interface – интерфейс «узел-сеть»

NNTR – Network News Transport Protokol – протокол передачи сетевых новостей

NRM – Normal Response Mode – режим нормального ответа

NRZ – Non Return to Zero – метод физического кодирования без возвращения к нулю

NRZI – Non Return to Zero with ones Inverted – потенциальный код с инверсией при единице

NTN – national terminal number – номер национального терминала

ODI – Open Data-Link Interface – открытый интерфейс передачи данных, если новый транспортный протокол разработан с использованием спецификаций ODI-интерфейса

ODP – Originator Detection Pattern – «шаблон обнаружения вызывающего» в протоколе LAPM

OLE – Object Linking and Pattern – объектная компановка

OMC – Operations and Maintenance Centre – центр управления и обслуживания

OSI – Open System Interconnection basic reference model – базовая эталонная модель взаимодействия открытых систем

OSPF – Open Shortest Path First – протокол маршрутизации стека TCP/IP, основанный на алгоритме состояния связей

PAD – Packet Assembler/Disassembler – сборщик/разборщик пакетов в сетях PSN

PAM – Pulse Amplitude Modulation – импульсная амплитудная модуляция

PCM – Pulse Code Modulation – импульсно-кодовая модуляция

PCMCIA – Personal Computer Memory Card International Association – Международная ассоциация производителей плат памяти для персональных компьютеров или одноименный стандарт на средства расширения портативных ПК

PCS – Phisical Coding Sublayer – подуровень кодирования в сетях Gigabit Ethernet

PDN – Public Data Network – сеть передачи данных общего пользования

PDU – Protocol Data Unit – протокольный блок данных

PEP – Packetized Ensemble Protocol – протокол пакетного кодирования в модемной связи (предложен фирмой Telebit)

PHY – Physical layer device – устройство физического уровня сетей Fast Ethernet

PMA – Physical Medium Attachment – средства подключения к физической среде

PMD – Physical Medium Dependent – подуровень, зависимый от физического носителя

PMI – Physical Medium Independent – подуровень, независимый от физического носителя

PO – Post Office – почтовое отделение

POP – Post Office Protocol – протокол обслуживания почтового офиса

PPP – Point-to-Point Protocol – протокол двухточечной связи

PRI – Primary Rate Interface – интерфейс первичного уровня в сетях ISDN

PRMD – Private management domain – частный управляющий домен

PS – Physical Signalling – подуровень физической сигнализации (ФС)

PSE – packet switching exchange – коммутатор переключения пакетов, входит в состав узла коммутации сетей PSN

PSN – Packet switching network – сеть с коммутацией пакетов

PSTN – Public Switched Telephone Network – коммутируемая телефонная сеть общего пользования (КТСОП)

PT – payload type – тип информационного наполнения, поле ячейки ATM, идентифицирующее тип данных

PTI – packet type identifier – «идентификатора типа пакета в заголовке 3-го уровня сети X.25

PVC – Permanent Virtual Circuit (channel) – постоянная(ый) виртуальная(ый) цепь (канал) или Permanent Virtual Connection – постоянное виртуальное соединение

QAM – Quadrature Amplitude Modulation – квадратурная амплитудная модуляция

RAID – Redundant Array of Independent Disk – матрица независимых дисковых накопителей, тип дисковой памяти с резервированием и дублированием

RDN – Relative Distinguished Name – относительное характерное имя

REJ – reject – неприем, кадр протокола HDLC

RFC – Request For Comments – запрос на комментарии и предложения (документ, в котором публикуются первоначальные варианты спецификаций при разработке стандартов, а также сами стандарты для интерсетей)

RIP – Routing Internet Protocol – протокол межсетевой маршрутизации стека TCP/IP

RLE – Run Length Encoding – групповое кодирование – алгоритм сжатия последовательности одинаковых символов, используемый, в частности, в растровой графике

RNR – Receive Not Ready – к приему не готов (кадр протокола HDLC)

RPC – Remonte Procedure Calls – протокол на основе удаленного вызова процедур

RR – receive ready – готов к приему (кадр протокола HDLC)

RSA – первые буквы фамилий авторов Rivest, Shamir, Adleman – алгоритм шифрования с общим ключом

SA – Source Address – адрес отправителя

SAP – Service Access Point – точка доступа к сервису. Используется для разделения входящего потока данных между приложениями на каждом из уровней, транспортном, сеансовом и представительном, или Service Advertising Protocol – протокол рекламного (сетевого) сервиса фирмы Novell, является надстройкой над протоколом IPX и используется файловым сервером для оповещения других станций о предоставляемых услугах

SCSI – Stall Computer System Interface – интерфейс малых вычислительных систем (стандарт)

SFD – Start Frame Delimiter – признак начала кадра

SFT – Server Fault Tolerance – отказоустойчивость сервера, или System Fault Tolerance – отказоустойчивость системы

SIM – Standard Identification Module – стандартный модуль подлинности абонента

SMP – Symmetric MultiProcessing – симметричная многопроцессорная обработка

SMS – Short Message Service – двунаправленная передача коротких сообщений (сервис в сетях сотовой подвижной связи), или Storage Management Services – служба управления запоминающими устройствами, стандартный набор API фирмы Novell для облегчения резервного копирования

SMTP – Simple Mail Transfer Protocol – простой протокол для обмена электронной почтой, или Simple Message Transfer Protocol – простой протокол передачи сообщений

SNA – System Network Architecture – системная сетевая архитектура (предложена фирмой IBM)

SNAP – Subnetwork Access Protocol – протокол доступа к подсетям, расширение формата кадра LLC

SNMP – Simple Network Management Protocol – упрощенный протокол управления сетью, простой протокол для мониторинга сети и сетевых компонентов

SNP – Signal Noise Pjwer – отношение мощности сигнала к мощности шума в канале

SPX – Seguenced Packet eXchange – протокол фирмы Novell, ориентированный на установление соединения

SREJ – selective reject – выборочный неприем, кадр протокола HDLC

SSAP – Source Service Access Point – точка доступа к сервису отправителя, определяет тип протокола сетевого уровня станции-отправителя в сетях Ethernet

SSB-SC – Single SideBand Suppressed Carrier – однополосная модуляция с подавленной несущей

SSL – Secure Socket Layer – протокол секретного обмена сообщениями

STP – ShieldedTwistedPair – экранированная витая пара

STX – Start of TeXt – символ начала текста, обычно определяет начало кадра в байт-ориенти-

рованных протоколах, в коде ASCII символ STX–02h имеет двоичное значение 00000010

SVC – Switched Virtual Circuit (channel) – коммутируемая(ый) виртуальная(ый) цепь (канал), или Switched Virtual Connection – коммутируемое виртуальное соединение

SYN – символ синхронизации в коде ASCII символ SYN–16h имеет двоичное значение 00010110

TCM – Trellis Coded Modulation – модуляция с решетчатым кодированием или треллис-кодированием

TCP – Transmission Control Protocol – протокол управления передачей с гарантированной доставкой данных, разбитых на последовательность фрагментов

TCP/IP – Transmission Control Protocol/Internet Protocol – протокол управления передачей/межсетевой протокол (стандарт, используемый в качестве протокола сетевого и транспортного уровня в Internet)

TCU – Trunk Coupling Unit – устройство подключения к магистрали в сетях Token Ring

TDM – Time Division Method – метод временного разделения

TDMA – Time Division Multiple Access – временное мультиплексирование с выделением слота по требованию (метод доступа в сетях сотовой подвижной связи)

THT – Token Holding Time – время удержания маркера станцией

TIA – Telecommunication Industry Assotiation – Ассоциация телекоммуникационной промышленности США

TLI – Transport Layer Protocol – протокол транспортного уровня, разработан фирмой AT&T и является надстройкой над протоколами IPX и SPX

TNC – Terminal Node Controller – контроллер конечного узла

TP – Transmission Path – пути или маршруты пересылки в ATM-сетях

TRT – Token Rotation Time – время оборота маркера (однократное прохождение маркера по логическому кольцу с учетом задержки доступа к среде передачи)

TTRT – Target Token Rotation Time – требуемое время оборота маркера, отражает степень потребности станции в пропускной способности кольца (чем меньше время TTRT, тем чаще станция желает получать маркер для передачи своих кадров)

UA – Unbalanced Asynchronous – симметричная конфигурация, или User Agent – пользовательский агент

UART – Universal Asynchronous Receiver Transmitter – универсальный асинхронный приемопередатчик

UDP – User Datagram Protocol – протокол дейтаграмм пользователя

UN – Unbalanced Normal – несбалансированная конфигурация

UNI – User Network Interface – интерфейс «абонент-сеть»

UPS – Uninterruptible Pover Supply – источник бесперебойного электропитания, или Uninterruptible Pover System – система бесперебойного электропитания

UTP – Unshielded Twisted pair – неэкранированная витая пара

VCC – Virtual Channel Connection – виртуальное соединение

VCI – virtual channel identifier – идентификатор виртуального канала в сетях ATM

VPI – virtual path identifier – идентификатор виртуального пути в сетях ATM

VSB – vestigial sideband – модуляция с частично подавленной боковой

WAN – work station – рабочая станция

WS – Wide Area Networks – глобальные сети

WWW – Wold-Wide-Web – всемирная «паутина» (сервисная глобальная гипертекстовая система в сети Internet)

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Учебное издание

ИНФОРМАТИКА В ТЕХНИЧЕСКОМ УНИВЕРСИТЕТЕ

Валерий Александрович Галкин
Юрий Александрович Григорьев

Телекоммуникации и сети

Редактор *Н.Е. Овчеренко*
Художники *О.В. Левашова, С.С. Водчиц*
Корректоры *Н.Л. Смирнова, М.А. Василевская*
Компьютерная верстка *И.Ю. Бурова*

Подписано в печать 06.11.02. Формат 70x100/16. Печать офсетная. Бумага офсетная.
Гарнитура «Таймс». Печ. л. 38. Усл. печ. л. 49,4. Уч.-изд. л. 48,92. Тираж 3000 экз.
Заказ 7220

Издательство МГТУ им. Н.Э. Баумана.
105005, Москва, 2-я Бауманская, 5.

Отпечатано с оригинал-макета в ГУП ППП «Типография «Наука».
121099, Москва, Шубинский пер., 6.