Министерство образования Республики Беларусь

Учреждение образования
«Полоцкий государственный университет»

Д. Г. Руголь

ВЫЧИСЛИТЕЛЬНЫЕ КОМПЛЕКСЫ, СИСТЕМЫ И СЕТИ

Учебно-методический комплекс для студентов специальностей
1-40 02 01 «Вычислительные машины, системы и сети»,
1-40 01 01 «Программное обеспечение информационных технологий»

Новополоцк
ПГУ
2014

УДК 004(075.8)
ББК 32.973я73
Р82

Рекомендовано к изданию методической комиссией
факультета информационных технологий
в качестве учебно-методического комплекса (протокол № 6 от 12.06.2013)

РЕЦЕНЗЕНТЫ:

начальник участка продажи услуг Полоцкого РУЭС
Витебского филиала РУП «Белтелеком» А. Ф. ЦУКАЛО;
канд. техн. наук, доц., зав. каф. вычислительных систем
и сетей УО «ПГУ» Р. П. БОГУШ

Руголь, Д. Г.

Р82 Вычислительные комплексы, системы и сети : учеб.-метод. комплекс для студентов специальностей 1-40 02 01 «Вычислительные машины, системы и сети», 1-40 01 01 «Программное обеспечение информационных технологий» / Д. Г. Руголь. – Новополоцк : ПГУ, 2014. – 348 с.

ISBN 978-985-531-458-6.

Рассмотрены варианты архитектур и принципы работы вычислительных комплексов, систем и сетей различного назначения. Приведена классификация вычислительных систем и сетей, особенности традиционных и перспективных технологий вычислительных сетей, основы их проектирования, способы создания крупных составных сетей и управления такими сетями, а также их характеристики, параметры и методы их анализа. Приведен обзор структурообразующего оборудования физического, канального и сетевого уровней вычислительных сетей, а также современных комплексов технических и программных средств, используемых в вычислительных сетях.

Предназначен для студентов специальностей 1-40 02 01 «Вычислительные машины, системы и сети» и 1-40 01 01 «Программное обеспечение информационных технологий».

УДК 004(075.8)
ББК 32.973я73

ISBN 978-985-531-458-6

ВВЕДЕНИЕ

Цель и задачи дисциплины

Основная цель курса «Вычислительные комплексы, системы и сети» – теоретическое и практическое изучение архитектур и принципов работы вычислительных комплексов, систем и сетей различного назначения, особенностей традиционных и перспективных технологий вычислительных сетей, основ их проектирования, способов создания крупных составных сетей и управления такими сетями.

Основные задачи курса:

- изучение принципов построения и функционирования вычислительных систем и сетей, их характеристик и параметров, а также методов их анализа;
- изучение структурообразующего оборудования физического, канального и сетевого уровней вычислительных сетей;
- изучение современных комплексов технических и программных средств, используемых в вычислительных сетях;
- наработка практического опыта по проектированию вычислительных сетей.

В результате изучения курса студенты должны:

- знать основные виды архитектур вычислительных комплексов, систем и сетей различного назначения;
- знать эталонную модель взаимодействия открытых систем;
- иметь представление о принципах построения, работы и взаимодействия основных устройств вычислительных систем в процессе их функционирования;
- знать логическую и физическую структуру компьютерных сетей;
- знать способы организации информационных связей в сетях;
- иметь представление об основных протоколах и интерфейсах в компьютерных сетях;
- знать методы повышения надежности и производительности вычислительных сетей;
- иметь представление о перспективах развития вычислительных комплексов, систем и сетей.

Структура дисциплины

Согласно учебному плану курс «Вычислительные комплексы, системы и сети» изучается студентами на 4 курсе (7, 8 семестры), рассчитан на 136 аудиторных часов и включает в себя следующие виды занятий:

- 80 часов лекций;
- 16 часов практических занятий;
- 40 часов лабораторных работ.

Ниже представлено распределение курса по видам аудиторных занятий по разделам и темам.

ЛЕКЦИОННЫЙ КУРС

№ п/п	Наименования разделов и тем лекций и их содержание	Количество часов
1	2	3
	VII семестр	
	Введение в курс «Вычислительные комплексы, системы и сети»	
1	Содержание дисциплины и ее взаимосвязь с другими дисциплинами	2
	Раздел I. Вычислительные комплексы и системы	**16**
2	Принципы параллельной обработки информации. Способы повышения быстродействия ЭВМ	2
3	Многомашинные вычислительные комплексы (ММВК). Классификация. Особенности программного обеспечения. Примеры ММВК	2
4	Многопроцесорные вычислительные комплексы (МПВК). Классификация. Специфика требований к программному обеспечению	2
5	Конвейерные вычислительные системы. Функционирование конвейера	2
6	Матричные вычислительные системы. Назначение и упрощенная структура матричной ВС. Функционирование процесорной матрицы	2
7	Ассоциативные и систолические вычислительные системы. Характер задач, решаемых при помощи ассоциативных и систолических ВС	2
8	Вычислительные системы с проблемно- и функционально ориентированными процессорами	2
	Раздел II. Компьютерные сети	**64**
9	Эволюция вычислительных систем. Основные программные и аппаратные компоненты сети. Современные тенденции	2
10	Понятие «открытая система». Модель OSI/ISO. Уровни модели OSI	2
11	Стандартные стеки коммуникационных протоколов. Стек OSI. Стек TCP/IP. Стек IPX/SPX. Стек NetBIOS/SMB	2
12	Линии связи. Типы линий связи. Аппаратура линий связи. Характеристики линий связи	2
13	Стандарты кабелей	2
14	Методы передачи дискретных данных на физическом уровне	2
15	Методы передачи данных канального уровня	2
16	Методы коммутации. Коммутация каналов	2
17	Коммутация пакетов. Принципы коммутации пакетов	2
18	Протоколы и стандарты локальных сетей. Структура стандартов IEEE 802.X	2
19	Протокол LLC уровня управления логическим каналом (802.2)	2
20	Технология Ethernet (802.3). Метод доступа CSMA/CD	2
	Итого:	**40**

VIII семестр		
1	2	3
21	Технология Ethernet**.** Форматы кадров технологии Ethernet. Стандарт 10Base-2	2
22	Технология Ethernet. Стандарты 10Base-5, 10Base-T, 10Base-F	2
23	Методика расчета конфигурации сети Ethernet	2
24	Технология Token Ring (802.5). Маркерный метод доступа к разделяемой среде	2
25	Технология FDDI. Основные характеристики. Физический уровень технологии. Технология 100VG-AnyLAN. Основные характеристики. Физический уровень технологии	2
26	Fast Ethernet как развитие технологии Ethernet. Физический уровень технологии. Стандарты 100Base-FX, 100Base-TX, 100Base-T4	2
27	Технология Gigabit Ethernet. Общая характеристика стандарта. Спецификации физической среды	2
28	Построение локальных сетей по стандартам физического и канального уровней. Структурированная кабельная система	2
29	Сетевые адаптеры. Функции и характеристики сетевых адаптеров. Концентраторы. Основные и дополнительные функции концентраторов	2
30	Логическая структуризация сети с помощью мостов и коммутаторов. Ограничения сети, построенной на общей разделяемой среде. Преимущества логической структуризации сети	2
31	Принципы работы мостов. Ограничения топологии сети, построенной на мостах	2
32	Коммутаторы локальных сетей	2
33	Техническая реализация коммутаторов	2
34	Характеристики, влияющие на производительность коммутаторов. Дополнительные функции коммутаторов	2
35	Виртуальные локальные сети. Типовые схемы применения коммутаторов в локальных сетях	2
36	Сетевой уровень как средство построения больших сетей. Принципы объединения сетей на основе протоколов сетевого уровня	2
37	Адресация в IP-сетях. Порядок распределения IP-адресов. Отображение IP-адресов на локальные адреса	2
38	Протокол IP. Основные функции протокола IP. Структура IP-пакета	2
39	Протоколы маршрутизации в IP-сетях	2
40	Основные характеристики маршрутизаторов. Стирание граней между коммутаторами и маршрутизаторами. Перспективы развития комплексов, систем и сетей	2
	Итого:	**40**
	ВСЕГО:	**80**

ЛАБОРАТОРНЫЕ ЗАНЯТИЯ

Наименование лабораторной работы	Количество часов
1. Windows Server 2003. Службы DHCP, WINS. Управление пользователями (разрешения, роли)	4
2. Windows Server 2003. Настройка DNS, создание и настройка дерева доменов	4
3. Windows Server 2003. Настройка IIS(HTTP, FTP). Служба маршрутизации, статические маршруты	4
4. Настройка аппаратного маршрутизатора	4
5. Windows Server 2003 Active Directory. Создание домена, настройка клиентов	4
6. Windows Server 2003 Active Directory. Создание дерева доменов	4
7. Ubuntu Server. Настройка службы DHCP	4
8. Ubuntu Server. Настройка службы DNS. Создание дерева доменов	4
9. Ubuntu Server. Настройка службы IPTABLES	4
10. Ubuntu Server. Настройка прокси-сервера SQUID	4
ВСЕГО:	**40**

ПРАКТИЧЕСКИЕ ЗАНЯТИЯ

Наименование практического занятия	Количество часов
1. Ubuntu Server. Команды файловой системы, редактирование файлов, управление пользователями, настройка сети	2
2. Ubuntu Server. Настройка пакета SAMBA	2
3. Server 2003. Служба VPN. Настройка VPN-сервера	2
4. Проектирование аппаратной. Проектирование кроссовой	2
5. Проектирование и расчет кабельной трассы подсистемы внутренних магистралей	2
6. Проектирование подсистемы рабочего места	2
7. Проектирование горизонтальной подсистемы. Проектирование точек перехода	2
8. Расчет декоративных коробов и монтажных конструктивов	2
ВСЕГО:	**16**

Оценка знаний студентов

Для оценки работы и знаний студентов в результате изучения курса «Вычислительные комплексы, системы и сети» используется накопительная система. Результирующая оценка выставляется по сумме баллов, которые студент набирает в течение всего учебного семестра, а также в результате выходного итогового контроля – экзамена.

Для получения аттестации необходимо выполнение всех предшествующих аттестации лабораторных работ.

Распределение баллов по видам занятий

Вид занятий	Форма оценки активности студента	Максимальное количество баллов по каждой форме оценки	Максимальное количество баллов по каждому виду занятий
Лабораторные занятия	Защита работы в течение отведенного на нее занятия	+5	25 × 4 = 100
	Защита работы в течение семестра	20	
Экзамен	Качество ответов на экзаменационные вопросы	100	100

Дополнительные баллы предусматриваются за выполнение задач повышенной сложности (до 100 баллов). Для получения аттестации студент должен сдать все предшествующие лабораторные работы.

Итоговая оценка выставляется по следующей шкале.

Оценка	1	2	3	4	5	6	7	8	9	10
Сумма баллов	0-79	80-99	100-119	120-139	140-159	160-179	180-189	190-199	200-219	220 и более

Для получения минимальной положительной оценки – 4 балла – студент должен набрать 120 баллов, для чего требуется:

- выполнить все лабораторные работы – минимум 80 баллов;
- получить 40 баллов при сдаче экзамена.

Для получения высшей оценки – 10 баллов – студенту необходимо будет проявить способность самостоятельно и творчески решать задачи повышенной сложности.

МОДУЛЬ 1. ВЫЧИСЛИТЕЛЬНЫЕ КОМПЛЕКСЫ И СИСТЕМЫ

Цель модуля – приобретение студентами общих понятий об организации и типах вычислительных систем, уровнях и способах организации обработки информации.

В результате изучения модуля студенты **должны:**

- иметь представление об архитектуре современных вычислительных систем;
- знать общую логическую организацию вычислительных систем;
- представлять назначение и принципы взаимодействия технических средств и программного обеспечения;
- уметь рационально выбирать структуру системы и способы организации вычислительных процессов.

Содержание модуля:

1.1. Принципы параллельной обработки информации.
1.2. Многомашинные вычислительные комплексы.
1.3. Многопроцессорные вычислительные комплексы.
1.4. Конвейерные вычислительные системы.
1.5. Матричные вычислительные системы.
1.6. Ассоциативные и систолические системы.
1.7. Функционально распределенные и перестраиваемые вычислительные системы.

Вопросы и задания для самопроверки.

1.1. Принципы параллельной обработки информации

Существуют два метода повышения быстродействия ЭВМ:

- совершенная элементная база;
- параллельное выполнение вычислительных операций.

Максимальная вычислительная мощность интегральной схемы определяется выражением

$$V_{\max} = fN / a ,$$

где f – частота переключения вентилей в микросхеме; N – количество вентилей; a – число переключений, необходимых для одной арифметико-логической операции.

Энергия переключения полупроводникового вентиля

$$E = \tau p,$$

где τ, p – время и мощность переключения вентиля;

$$fN = D / E,$$

где D – допустимая числовая мощность, рассеиваемая микросхемой.

Данные свидетельствуют, что могут быть получены достаточно большие значения быстродействия. Однако функционально СБИС и БИС реализуют принцип действия последовательных ЭВМ, в результате задействована малая часть вентилей и $V_{ПС} << V_{\max}$. Следовательно, для увеличения вычислительной мощности ЭВМ необходимо научиться использовать суммарную мощность ИС, т. е. применять параллелизм.

Параллелизм – это возможность одновременного выполнения нескольких арифметико-логических или служебных операций. На стадии постановки задачи параллелизм не определен, он появляется только после выбора метода вычисления. Зависит от метода и уровня подготовки пользователя алгоритма. Параллелизм используемой ЭВМ также меняется в широких пределах и зависит, в первую очередь, от числа процессоров, способа размещения данных методов коммутации и синхронизации процессов. Язык программирования является средством переноса параллелизма алгоритма на параллелизм ЭВМ, возможности языка также влияют на результат переноса.

Одновременное выполнение операций возможно, если они логически независимы.

Таким образом, описание зависимостей операций по данным полностью определяют параллелизм метода вычисления.

Программные объекты A и B (команды, операторы, программы) являются независимыми и могут выполняться параллельно, если выполняется следующее условие:

$$[In(B) \wedge Out(A)] \vee [In(A) \wedge Out(B)] \vee [Out(A) \wedge Out(B)] = null,$$

где $In(A)$ и $Out(A)$ – набор входных и выходных данных для объекта A, аналогично – $In(B)$ и $Out(B)$.

Нарушение условия в первом терме – прямая зависимость между программными объектами:

$A: R = R_1 + R_2$

$B: Y = R + X$

Здесь операторы A и B не могут выполняться одновременно, т. к. результат A является операндом B.

Нарушение условия во втором терме – обратная зависимость:

$A: R = R_1 + R_2$

$B: R_1 = X + Y$

Здесь выполнение B изменяет операнд A.

Нарушение условия в третьем терме – конкуренция:

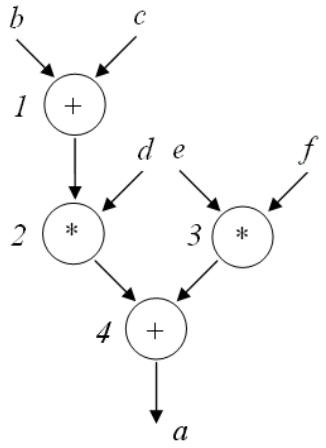
$A: R = A + B$

$B: R = X + Y$

Здесь выполнение B и A дают неопределенный результат.

Увеличение параллелизма в любой задаче заключается в поиске и устранении таких зависимостей.

Общей формой представления перечисленных зависимостей является *информационный граф* (ИГ) задачи (рис. 1.1).



a=(b+c)d+ef

1	+	*b*	*c*
2	*	~	*d*
3	*	*e*	*f*
4	+	~	~

Рис. 1.1. Информационный граф математического выражения

Более определенная форма представления параллелизма – *ярусно-параллельная форма* (ЯПФ). В нулевой ярус входят операторы (ветви), не зависящие друг от друга, в первый ярус – зависящие только от второго, и т. д.

Параметры ЯПФ:

b_i – ширина яруса i; B – ширина графа ЯПФ (максимум b_i); l_i – длина яруса; L – длина графа; ε – коэффициент заполнения ярусов; θ – коэффициент разброса указанных параметров.

Формы параллелизма. Можно выделить следующие формы параллелизма:

- естественный или векторный параллелизм;
- параллелизм независимых ветвей;
- параллелизм смежных операций или скалярный параллелизм.

Векторный параллелизм. Наиболее распространенная форма параллелизма. Вектор – одномерный массив (образуется из многомерного массива, когда один его индекс пробегает все значения в диапазоне своего изменения, в параллельных языках обозначается *). Векторная операция:

C(*, j) = A(*, j) + B(*, j).

На последовательном языке:

DO 1 I = 1, N;

1 C(I, J) = A(I, J) + B(I, J).

Параллелизм независимых ветвей. Выделение в программе независимых алгоритмических структур – ветвей. В параллельном языке запуск ветви выполняется оператором FORK M1, M2, M3, … где MX – имена независимых ветвей. В конце каждой ветви стоит оператор JOIN(R, K); R – блок памяти, хранящий число ветвей и декрементирующийся после вызова JOIN. Когда он становится равным нулю, то происходит переход на выполнение по метке K.

Скалярный параллелизм. При выполнении программы постоянно встречаются ситуации, когда исходные данные для I-й операции вычисляются на предстоящих стадиях, и при определенной форме вычислительной системы их можно совместить.

Дана формула (расчет ширины запрещенной зоны):

$$E = \frac{mq^4\pi^2}{8\varepsilon_0^2\varepsilon^2 h^2}.$$

Последовательная программа:

E = (m · q^4 · p^2) / (8 · e0^2 · e^2 · h^2).

На рис. 1.2. представлен граф, соответствующий вышеприведенной формуле.

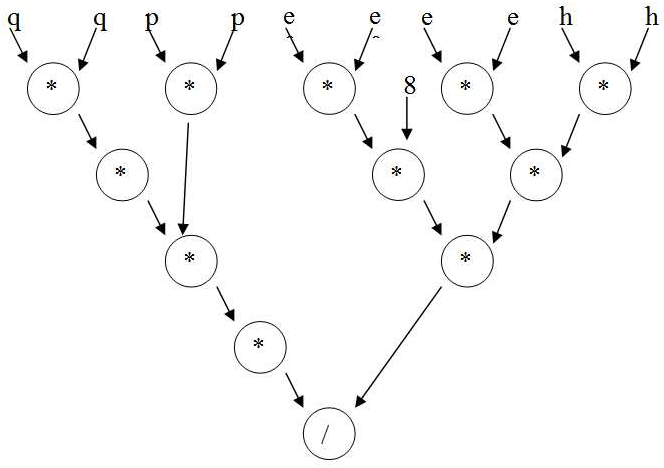


Рис. 1.2. Ярусно-параллельная форма представления параллелизма

Для столь элементарных операций параллелизм должен обеспечиваться аппаратурой ЭВМ.

Ускорение параллельных ЭВМ

Ускорение используется на этапе проектирования или в научных исследованиях для оценки предельных возможностей архитектуры:

$$r = T_1 / T_n,$$

где T_1 – время решения задачи на однопроцессорной системе; T_n – время решения задачи на *n*-процессорной системе.

Если

$$W = W_{ск} + W_{пр},$$

где W – общее число операций в задаче; $W_{ск}$ – число скалярных операций; $W_{пр}$ – число операций, которые могут выполняться параллельно,

то (*закон Амдала*) $r = \dfrac{Wt}{(W_{ск} + \dfrac{W_{пр}}{n})t} = \dfrac{1}{a + \dfrac{1-a}{n}} \xrightarrow[n\to\infty]{} \dfrac{1}{a}$,

где t – время выполнения одной операции; $a = \dfrac{W_{ск}}{W}$ – удельный вес скалярных операций.

Из закона следует:

- ускорение зависит как от потенциального параллелизма задачи, так и от параметров аппаратуры;
- предельное ускорение определяется свойствами задачи.

В реальных параллельных системах существует несколько уровней параллелизма, и для каждого уровня определено свое ускорение. Для полного ускорения ЭВМ используют формулу $R = \prod_{i=1}^{P} r_i$, где P – число уровней вычислений; r_i – ускорение на уровне i.

1.2. Многомашинные вычислительные комплексы

Многомашинный вычислительный комплекс (ММВК) – комплекс, включающий в себя две или более ЭВМ (каждая из которых имеет процессор, ОЗУ, набор периферийных устройств и работает под управлением собственной операционной системы), связи между которыми обеспечивают выполнение функций, возложенных на комплекс.

Цели, которые ставятся при объединении ЭВМ в комплекс, могут быть различными, и они определяют характер связей между ЭВМ. Чаще всего основной целью создания ММВК является или увеличение производительности, или повышение надежности, или одновременно и то, и другое. Однако при достижении одних и тех же целей связи между ЭВМ могут существенно различаться.

По характеру связей между ЭВМ комплексы можно разделить на три типа: ***косвенно-, или слабосвязанные***; ***прямосвязанные***; ***сателлитные***.

В ***косвенно-,*** или ***слабосвязанных,*** комплексах ЭВМ связаны друг с другом только через внешние запоминающие устройства (ВЗУ). Для обеспечения таких связей используются устройства управления ВЗУ с двумя и более входами. Структурная схема такого ММВК приведена на рис. 1.3. Заметим, что здесь и далее для простоты приводятся схемы для двухмашинных комплексов. При трех и более ЭВМ комплексы строятся аналогичным образом. В косвенно связанных комплексах связь между ЭВМ осуществляется только на информационном уровне. Обмен информацией осуществляется в основном по принципу «почтового ящика», т. е. каждая из ЭВМ помещает в общую внешнюю память информацию, руководствуясь собственной программой, и соответственно, другая ЭВМ принимает эту информацию, исходя из своих потребностей.

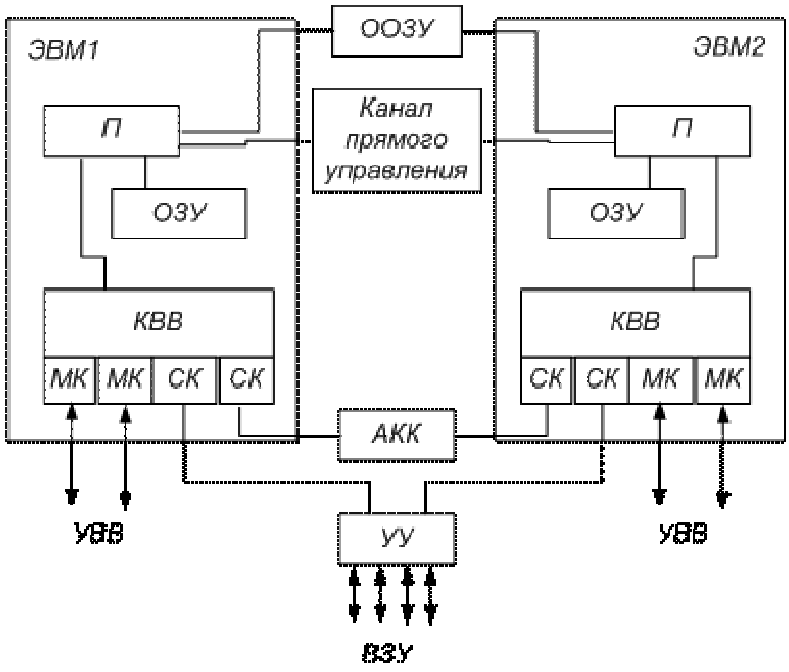


Рис. 1.3. Связи между ЭВМ и ММВК

Такая организация связей обычно используется в тех случаях, когда ставится задача повысить надежность комплекса путем резервирования ЭВМ. В этом случае ЭВМ, являющаяся основной, решает заданные задачи, выдает результаты и постоянно оставляет в общем ВЗУ всю информацию, необходимую для продолжения решения с любого момента времени. Вторая ЭВМ, являющаяся резервной, может находиться в состоянии ожидания, с тем, чтобы в случае выхода из строя основной ЭВМ по сигналу оператора начать выполнение функций, используя информацию, хранимую в общем ВЗУ основной ЭВМ.

При такой связи может быть несколько способов организации работы комплекса:

1. Резервная ЭВМ находится в выключенном состоянии (ненагруженный резерв) и включается только при отказе основной ЭВМ. Естественно, для того чтобы резервная ЭВМ начала выдавать результаты вместо основной, потребуется определенное время, которое определяется временем, необходимым для включения ЭВМ, вхождением ее в режим, а также временем, отводимым для проверки ее исправности. Это время может быть достаточно большим. Такая организация возможна, когда система, в которой работает ЭВМ, не критична по отношению к некоторым перерывам или остановкам в процессе решения задач. Это обычно имеет место в случаях, когда ЭВМ не выдает управляющую информацию.

2. Резервная ЭВМ находится в состоянии полной готовности и в любой момент может заменить основную ЭВМ (нагруженный резерв), причем либо не решает никаких задач, либо работает в режиме самоконтроля, решая контрольные задачи. В этом случае переход в работе от основной к резервной ЭВМ может осуществляться достаточно быстро, практически без перерыва в выдаче результатов. Однако следует заметить, что основная ЭВМ обновляет в общем ВЗУ информацию, необходимую для продолжения решения, не непрерывно, а с определенной дискретностью, поэтому резервная ЭВМ начинает решать задачи, возвращаясь на некоторое время назад. Такая организация допустима и в тех случаях, когда ЭВМ работает непосредственно в контуре управления, а управляемый процесс достаточно медленный и возврат во времени не оказывает заметного влияния.

При организации работы по первому и второму вариантам ЭВМ используются нерационально: одна ЭВМ всегда простаивает. Простоев можно избежать, загружая ЭВМ решением каких-то вспомогательных задач, не имеющих отношения к основному процессу. Это повышает эффективность системы – производительность практически удваивается.

3. Для того чтобы полностью исключить перерыв в выдаче результатов, обе ЭВМ, и основная, и резервная, решают одновременно одни и те же задачи, но результаты выдаст только основная ЭВМ, а в случае выхода ее из строя результаты начинает выдавать резервная ЭВМ. При этом общее ВЗУ используется только для взаимного контроля. Иногда такой комплекс дополняется устройством для сравнения результатов с целью контроля. Если при этом используются три ЭВМ, то возможно применение метода голосования, когда окончательный результат выдается только при совпадении результатов решения задачи не менее чем от двух ЭВМ. Это повышает и надежность комплекса в целом, и достоверность выдаваемых результатов. Разумеется, в этом варианте высокая надежность и оперативность достигается весьма высокой ценой – увеличением стоимости системы.

Следует обратить внимание, что при любой организации работы и слабосвязанном ММВК переключение ЭВМ осуществляется либо по командам оператора, либо с помощью дополнительных средств, осуществляющих контроль исправности ЭВМ и вырабатывающих необходимые сигналы. Кроме того, быстрый переход к работе с основной на резервную ЭВМ возможен лишь при низкой эффективности использования оборудования.

Существенно большей гибкостью обладают прямосвязанные ММВК. В прямосвязанных комплексах существуют три вида связей (см. рис. 1.3): общее ОЗУ (ООЗУ); прямое управление, иначе – связь процессор – процессор; адаптер канал – канал (АКК).

Связь через общее ОЗУ гораздо сильнее связи через ВЗУ. Хотя первая связь также носит характер информационной связи и обмен информацией осуществляется по принципу «почтового ящика», однако, вследствие того, что процессоры имеют прямой доступ к ОЗУ, все процессы в системе могут протекать с существенно большей скоростью, а разрывы в выдаче результатов при переходах с основной ЭВМ на резервную сокращаются до минимума. Недостаток связи через общее ОЗУ заключается в том, что при выходе из строя ОЗУ, которое является сложным электронным устройством, нарушается работа всей системы. Чтобы этого избежать, приходится строить общее ОЗУ из нескольких модулей и резервировать информацию. Это, в свою очередь, приводит к усложнению организации вычислительного процесса в комплексе и в конечном счете – к усложнению операционных систем. Следует отметить также и то, что связи через общее ОЗУ существенно дороже, чем через ВЗУ.

Непосредственная связь между процессорами – канал прямого управления – может быть не только информационной, но и командной, т. е. по каналу прямого управления один процессор может непосредственно управлять действиями другого процессора. Это, естественно, улучшает динамику перехода от основной ЭВМ к резервной, позволяет осуществлять более полный взаимный контроль ЭВМ. Вместе с тем, передача сколько-нибудь значительных объемов информации по каналу прямого управления нецелесообразна, так как в этом случае решение задач прекращается: процессоры ведут обмен информацией.

Связь через адаптер канал – канал в значительной степени устраняет недостатки связи через общее ОЗУ и вместе с тем почти не уменьшает возможностей по обмену информацией между ЭВМ по сравнению с общим ОЗУ. Сущность этого способа связи заключается в том, что связываются между собой каналы двух ЭВМ с помощью специального устройства – адаптера. Обычно это устройство подключается к селекторным каналам ЭВМ. Такое подключение адаптера обеспечивает достаточно быстрый обмен информацией между ЭВМ, при этом обмен может производиться большими массивами информации. В отношении скорости передачи информации связь через АКК мало уступает связи через общее ОЗУ, а в отношении объема передаваемой информации – связи через общее ВЗУ. Функции АКК достаточно просты: это устройство должно обеспечивать взаимную синхронизацию работы двух ЭВМ и буферизацию информации при ее передаче. Хотя функции АКК и его структура (см. рис. 1.3) достаточно просты, однако большое разнообразие режимов работы двух ЭВМ и необходимость реализации этих режимов существенно усложняют это устройство.

Прямосвязанные комплексы позволяют осуществлять все способы организации ММВК, характерные для слабосвязанных комплексов. Однако за счет некоторого усложнения связей эффективность комплексов может быть значительно повышена. В частности, в прямосвязанных комплексах возможен быстрый переход от основной ЭВМ к резервной и в тех случаях, когда резервная ЭВМ загружена собственными задачами. Это позволяет обеспечивать высокую надежность при высокой производительности.

В реальных комплексах одновременно используется не один вид связи между ЭВМ, а два или более. Очень часто в прямосвязанных комплексах присутствует и косвенная связь через ВЗУ.

Для комплексов с сателлитными ЭВМ характерным является не способ связи, а принципы взаимодействия ЭВМ. Структура связей в сателлитных комплексах не отличается от связей в обычных ММВК: чаще всего

связь между ЭВМ осуществляется через АКК. Особенностью же этих комплексов является то, что в них, во-первых, ЭВМ существенно различаются по своим характеристикам, а во-вторых, имеет место определенная соподчиненность машин и различие функций, выполняемых каждой ЭВМ. Одна из ЭВМ, основная, является, как правило, высокопроизводительной и предназначается для основной обработки информации. Вторая, существенно меньшая по производительности, называется сателлитной, или вспомогательной, ЭВМ. Ее назначение – организация обмена информацией основной ЭВМ с периферийными устройствами, ВЗУ, удаленными абонентами, подключенными через аппаратуру передачи данных к основной ЭВМ. Кроме того, сателлитная ЭВМ может производить предварительную сортировку информации, преобразование ее в форму, удобную для обработки на основной ЭВМ, приведение выходной информации к виду, удобному для пользователя, и др. Сателлитная ЭВМ, таким образом, избавляет основную высокопроизводительную ЭВМ от выполнения многочисленных действий, которые не требуют ни большой разрядности, ни сложных операций, т. е. операций, для которых большая, мощная ЭВМ не нужна. Более того, с учетом характера выполняемых сателлитной машиной операций она может быть ориентирована на выполнение именно такого класса операций и обеспечивать даже бóльшую производительность, чем основная ЭВМ.

Некоторые комплексы включают в себя не одну, а несколько сателлитных ЭВМ, при этом каждая из них ориентируется на выполнение определенных функций, например, одна осуществляет связь основной ЭВМ с устройствами ввода – вывода информации, другая – связь с удаленными абонентами, третья организует файловую систему и т. д.

Появление в последнее время дешевых и простых микро-ЭВМ в немалой степени способствует развитию сателлитных комплексов. Сателлитные комплексы решают только одну задачу: увеличивают производительность комплекса, не оказывая заметного влияния на показатели надежности.

Подключение сателлитных ЭВМ принципиально возможно не только через АКК, но и другими способами, однако связь через АКК наиболее удобна.

1.3. Многопроцессорные вычислительные комплексы

Многопроцессорный вычислительный комплекс (МПВК) – это комплекс, включающий в себя два или более процессоров, имеющих общую оперативную память, общие периферийные устройства и работающих под

управлением единой операционной системы (ОС), которая, в свою очередь, осуществляет общее управление техническими и программными средствами комплекса. Следует оговорить, что каждый из процессоров может иметь индивидуальные, доступные только ему ОЗУ и периферийные устройства. Все перечисленное весьма существенно, так как делает возможной гибкую организацию параллельной обработки информации и позволяет наиболее эффективно использовать все ресурсы комплекса.

На рис. 1.4 представлена упрощенная схема МПВК, содержащая три процессора, два модуля ОЗУ и одну подсистему ввода – вывода информации (ПВВ). Даже для такого простого варианта схема оказывается достаточно сложной, так как в МПВК должен быть обеспечен доступ любого процессора и любого канала ввода – вывода к любой ячейке ОЗУ, любого процессора к любому каналу и периферийному устройству. Если представить теперь, что процессоров существенно больше, что ОЗУ по соображениям надежности и удобства наращивания емкости выполнено в виде нескольких модулей, а подсистема ввода – вывода включает в себе несколько каналов и большое число периферийных устройств, то становится ясным, насколько сложна топология МПВК. Если же учесть то обстоятельство, что для работы ОС аппаратные средства должны обеспечивать работу с переменными логическими адресами ОЗУ, периферийных устройств и каналов ввода – вывода, защиту памяти от взаимного влияния различных программ и возможность запуска одним процессором другого, сложность аппаратной реализации МПВК становится ясной в полной мере.

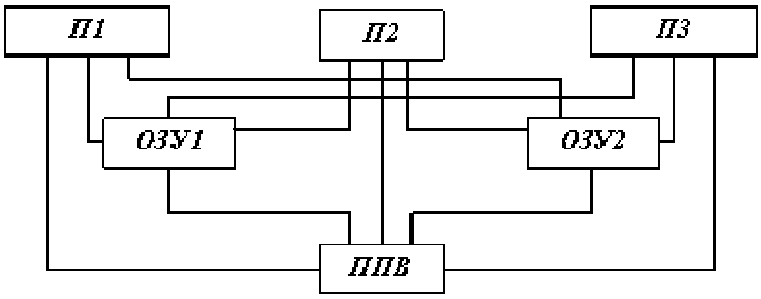


Рис. 1.4. Связи в МПВК

Непростые задачи возникают и при организации вычислительного процесса в МПВК., т. е. при построении ОС, которые являются основным средством организации всех процессов обработки информации в комплексе. Кроме обычных функций, выполняемых ОС при мультипрограммной обработке информации, возникают такие задачи, как распределение ресурсов и заданий между процессорами, синхронизация процессов при реше-

нии несколькими процессорами одной задачи, планирование с учетом оптимизации загрузки всех процессоров. При этом надо иметь в виду, что в процессе работы в комплексе возникает большое число конфликтных ситуаций, которые должны обрабатываться ОС. Эти и ряд других обстоятельств и факторов, связанных с обеспечением высокой надежности, делают ОС МПВК чрезвычайно сложной.

Однако, несмотря на все трудности, связанные с аппаратной и программной реализацией, МПВК получают все большее распространение, так как обладают рядом достоинств, основные из которых:

- высокая надежность и готовность за счет резервирования и возможности реконфигурации;
- высокая производительность за счет возможности гибкой организации параллельной обработки информации и более полной загрузки всего оборудования;
- высокая экономическая эффективность за счет повышения коэффициента использования оборудования комплекса.

Рассматривая процесс появления и развития МПВК, по-видимому, следует признать, что первоначально перед МПВК ставилась только задача обеспечения высокой надежности системы.

Не случайно поэтому, что одним из первых (а может быть, и самым первым) МПВК был комплекс D-825, созданный фирмой «Барроуз» (США) в 1968 г. для систем военного назначения. Комплекс включал в себя 4 процессора, 16 модулей ОЗУ, 10 каналов ввода – вывода и до 256 периферийных устройств, т. е. был весьма представительным МПВК, по тем меркам. Надо сказать, что эта первая попытка была весьма удачной – поставленные задачи были полностью решены. Вместе с тем создание первых же МПВК выявило и возможности достижения с их помощью высокой производительности. В настоящее время МПВК чаще создаются именно с такой целью.

Типы структурной организации МПВК

Существуют три типа структурной организации МПВК: с *общей шиной*; с *перекрестной коммутацией*; с *многовходовыми ОЗУ*.

В комплексах *с общей шиной* проблема связей всех устройств между собой решается крайне просто: все они соединяются общей шиной, выполненной в виде совокупности проводов или кабелей, по которым передаются информация, адреса и сигналы управления (рис. 1.5, *а*). Интерфейс является односвязным, т. е. обмен информацией в любой момент времени может происходить только между двумя устройствами. Если потребность в

обмене существует более чем у двух устройств, то возникает конфликтная ситуация, которая разрешается с помощью системы приоритетов и организации очередей в соответствии с этим. Обычно функции арбитра выполняет либо процессор, либо специальное устройство, которое регистрирует все обращения к общей шине и распределяет шину во времени между всеми устройствами комплекса.

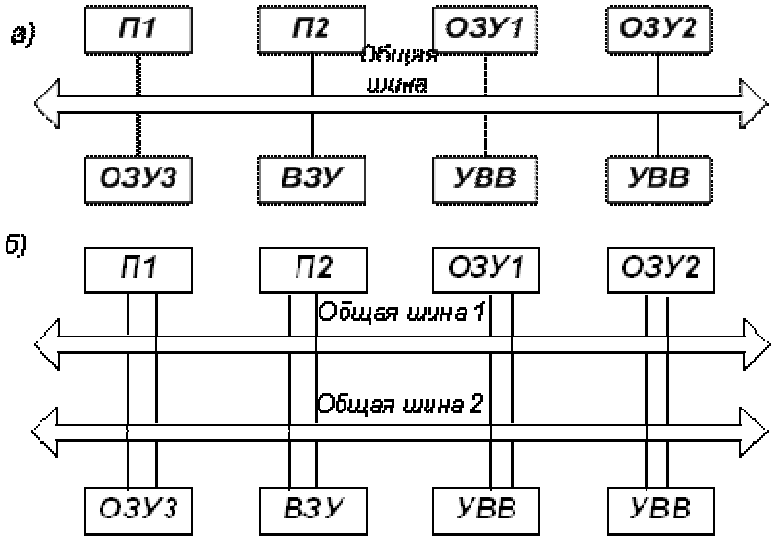


Рис. 1.5. МПВК с общей шиной

Несомненные достоинства структуры с общей шиной – простота, в том числе изменения комплекса, добавления или изъятия отдельных устройств, а также доступность модулей ОЗУ для всех остальных устройств. Следствием всего этого является достаточно низкая стоимость комплекса.

Вместе с тем комплексы с общей шиной не лишены определенных недостатков. Первый – невысокое быстродействие, так как одновременный обмен информацией возможен между двумя устройствами, не более. По этой причине в комплексах с общей шиной число процессоров не превосходит двух – четырех. Этот недостаток может быть несколько компенсирован путем использования общей шины с высоким быстродействием, бóльшим, чем быстродействие входящих в комплекс устройств. Однако этот путь приводит к усложнению и удорожанию комплекса. Второй недостаток МПВК с общей шиной заключается в относительно низкой надежности системы из-за наличия общего элемента – шины. Надо иметь в виду, что надежность общей шины определяется не только надежностью проводов и кабелей (их собственная надежность достаточно высока), но и надежностью всех соединений, входных и выходных цепей устройства. Отказ хотя бы одного из элементов приводит к отказу всего комплекса. Этот недостаток можно компенсировать за счет введения резервной шины

(см. рис. 1.5, *б*). Хотя это несколько усложняет комплекс, однако надежность его существенно возрастает. Если же резервную шину сделать активной, т. е. работающей одновременно с основной, то можно не только повысить надежность, но и увеличить производительность комплекса за счет того, что обмен информацией может осуществляться одновременно между двумя парами устройств.

Общая шина может быть организована различными способами – принципиально так же, как и для однопроцессорных ЭВМ с общей шиной.

Полностью лишены недостатков, присущих МПВК с общей шиной, МПВК *с перекрестной коммутацией*. Идея структурной организации таких ВК заключается в том, что все связи между устройствами осуществляются с помощью специального устройства – коммутационной матрицы (рис. 1.6). Коммутационная матрица (КМ) позволяет связывать друг с другом любую пару устройств, причем таких пар может быть сколько угодно – связи не зависят друг от друга.

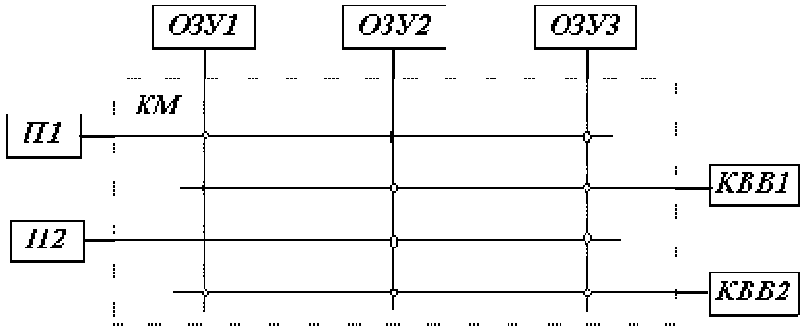


Рис. 1.6. МПВК с перекрестной коммутацией на основе коммутационной матрицы

В МПВК с перекрестной коммутацией нет конфликтов из-за связей, остаются только конфликты из-за ресурсов. Возможность одновременной связи нескольких пар устройств позволяет добиваться очень высокой производительности комплекса. Важно отметить и такое обстоятельство, как возможность установления связи между устройствами на любое, даже на длительное время, так как это совершенно не мешает работе других устройств, зато позволяет передавать любые массивы информации с высокой скоростью, что также способствует повышению производительности комплекса. Заметим, что в МПВК с общей шиной передача информации массивами, т. е. занятие шины одной парой устройств на длительный отрезок времени, обычно допускается лишь в крайних случаях, так как это приводит к длительным простоям остальных устройств.

Кроме того, к достоинствам структуры с перекрестной коммутацией можно отнести простоту и унифицированность интерфейсов всех уст-

ройств, а также возможность разрешения всех конфликтов в коммутационной матрице. Важно отметить и то, что нарушение какой-то связи приводит не к выходу из строя всего комплекса, а лишь к отключению какого-либо устройства, т. е. надежность таких комплексов достаточно высока. Однако и организация МПВК с перекрестной коммутацией не свободна от недостатков.

Прежде всего – сложность наращивания ВК. Если в коммутационной матрице заранее не предусмотреть большого числа входов, то введение дополнительных устройств в комплекс потребует установки новой коммутационной матрицы. Существенным недостатком является и то, что коммутационная матрица при большом числе устройств в комплексе становится сложной, громоздкой и достаточно дорогостоящей. (Надо учитывать то обстоятельство, что коммутационные матрицы строятся обычно на схемах, быстродействие которых существенно выше быстродействия схем и элементов основных устройств, – только при этом условии реализуются все преимущества коммутационной матрицы). Это обстоятельство в значительной степени усложняет и удорожает комплексы.

Для того чтобы упростить и удешевить ВК, коммутацию устройств осуществляют с помощью двух и даже более коммутационных матриц. На рис. 1.7 представлен МПВК, включающий в себя две матрицы: КМЦУ – матрицу для центральных устройств (процессоров, ОЗУ и каналов ввода – вывода) и КМПУ – матрицу для периферийных устройств. Схемы последней могут иметь существенно меньшее быстродействие, чем схемы первой, да к тому же обе коммутационные матрицы будут значительно проще и дешевле, чем одна общая коммутационная матрица с высоким быстродействием.

Перекрестная коммутация довольно широко используется при построении ВК, в частности, практически всех МПВК фирмы «Барроуз» (в том числе и упомянутого выше комплекса D-825).

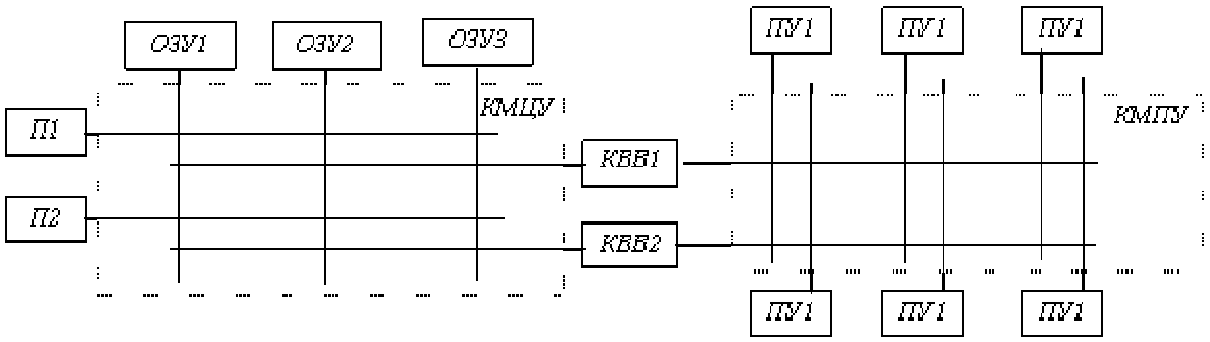


Рис. 1.7. МПВК с перекрестной коммутацией на раздельных коммутационных матрицах для центральных и периферийных устройств

В МПВК *с многовходовыми ОЗУ* все, что связано с коммутацией устройств, осуществляется в ОЗУ. В этом случае модули ОЗУ имеют число входов, равное числу устройств, которые к ним подключаются, т. е. для каждого устройства предусматривается свой вход в ОЗУ. Структура такого МПВК показана на рис. 1.8, *а*.

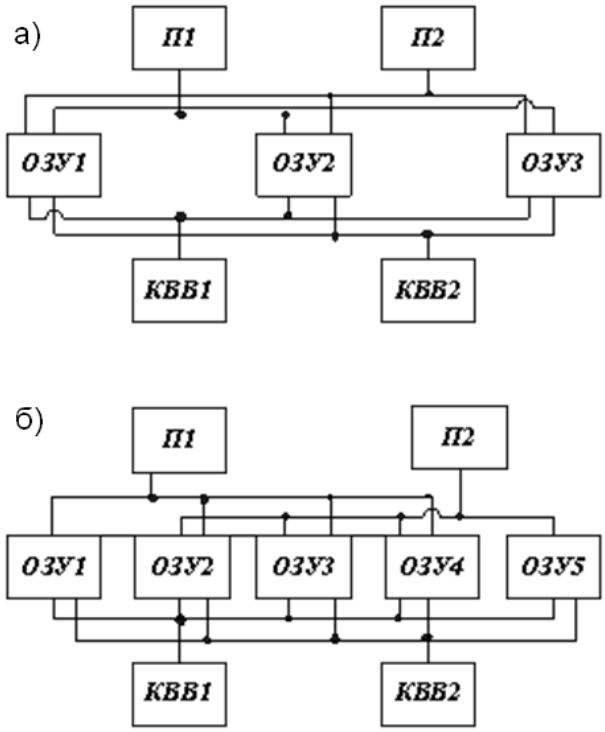


Рис. 1.8. МПВК с многовходовым ОЗУ

В отличие от ВК с перекрестной коммутацией, которые имеют централизованное коммутационное устройство, в МПВК с многовходовыми ОЗУ средства коммутации распределены между несколькими устройствами. Такой способ организации МПВК сохраняет все преимущества систем с перекрестной коммутацией, несколько упрощая при этом саму систему коммутации. Для наращивания системы должны быть предусмотрены дополнительные входы в ОЗУ. Правда, введение дополнительных модулей ОЗУ не вызывает затруднений.

В МПВК с многовходовыми ОЗУ очень просто решается вопрос о выделении каждому процессору своей оперативной памяти, недоступной другим процессорам. Такая организация показана на рис. 1.8, *б*. Выделение индивидуальной памяти каждому процессору позволяет хранить в ней информацию, которая необходима только одному процессору – различные таблицы и данные, копии некоторых модулей операционной системы и др. Это позволяет избежать части конфликтов, которые неизбежно возникают при общей оперативной памяти. Кроме того, уменьшается вероятность ис-

кажения информации в ОЗУ другими процессорами. Однако такие БК имеют тот недостаток, что в случае выхода из строя какого-либо процессора доступ к его памяти затруднен и информация может быть переписана в другой модуль ОЗУ только через канал ввода – вывода и внешнее ЗУ, что требует много времени.

Приведенные три типа структурной организации исчерпывают существующие построения МПВК, полностью удовлетворяющих тому определению, которое дано в начале этого пункта. Такие МПВК в литературе называют классическими, или истинными, МПВК.

Вместе с тем нередко МПВК называют комплексы, лишь частично удовлетворяющие этому определению, например, любые комплексы, в которых имеется несколько процессоров, а иногда даже матричные и конвейерные, которые (в отличие от классических МПВК) не относятся к комплексам с множественным потоком команд и множественным потоком данных. Кроме того, иногда к МПВК относят ЭВМ, имеющие не одно общее арифметическо-логическое устройство, а несколько операционных устройств (ОУ), каждое из которых выполняет определенную группу операций и может работать совершенно независимо от других. Таким образом, на первый взгляд может показаться, что такой комплекс имеет несколько процессоров, каждый из которых связан с общим ОЗУ. Однако все ОУ работают под управлением одной программы.

На рис. 1.9 приведен пример комплекса ЕС-1065, в котором применяются множественные ОУ. Его с полным правом можно назвать МПВК, так как кроме этих ОУ имеются два процессора команд (ПК), причем каждый может работать по собственной программе.

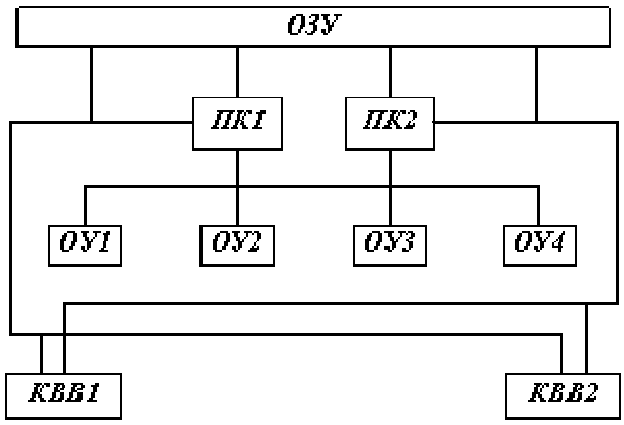


Рис. 1.9. Комплекс ЕС-1065

В этом комплексе налицо все элементы – несколько независимых процессоров, работающих с общедоступными ОЗУ, имеющих общие периферийные устройства и работающих под управлением общей операционной системы. По типу структурной организации этот комплекс может быть отнесен к МПВК с многовходовыми ОЗУ.

Нередко к МПВК относят комплексы, являющиеся по существу ММВК, но в которых для достижения более высокой надежности резервирование осуществляется не «помашинно», а по устройствам. Структурная схема одного из таких ВК приведена на рис. 1.10.

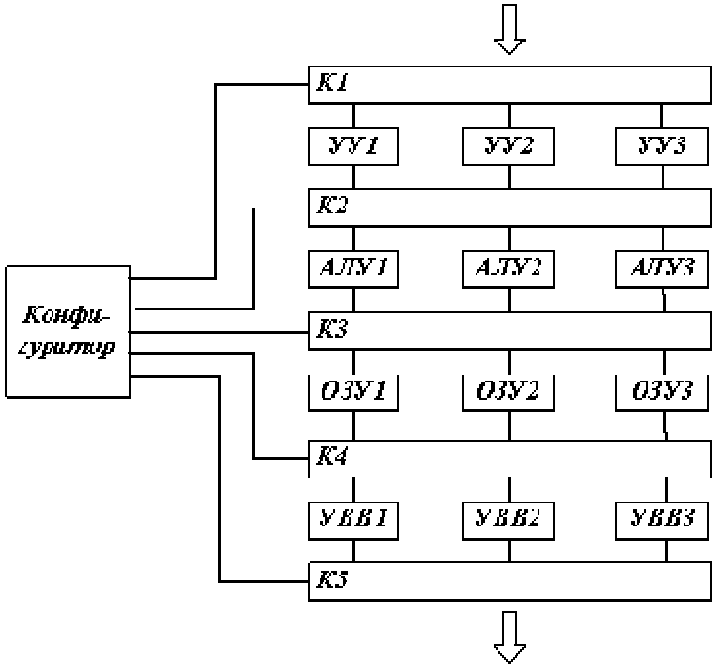


Рис. 1.10. Комплекс повышенной надежности

Комплекс первоначально работает как трехмашинный со сравнением результатов на выходе. Результат используется по методу «голосования». При несовпадении результата с помощью системы контроля определяется и отключается неисправное устройство. Эти функции выполняются конфигуратором, контролирующим состояние комплекса. Такой комплекс обладает высокой надежностью, однако, как видим, несмотря на наличие трех процессоров, его нельзя назвать многопроцессорным.

1.4. Конвейерные вычислительные системы

Одним из самых простых и наиболее распространенных способов повышения быстродействия процессоров является конвейеризация процесса вычислений. Большим преимуществом конвейерных ЭВМ перед парал-

лельными ЭВМ других типов является возможность использования пакетов программ, уже написанных для последовательных ЭВМ.

В любом процессоре машинная команда проходит ряд этапов обработки, например: выборку команды из оперативной памяти (ВК), вычисление абсолютного адреса операнда в оперативной памяти (ВА), выборку операнда из памяти (ВО), операцию в АЛУ.

В процессоре последовательной ЭВМ для выполнения этих функций используется единственное устройство, поэтому время выполнения команды определяется как

$$t_к = t_{ВК} + t_{ВА} + t_{ВО} + t_{АЛУ}.$$

Чтобы уменьшить $t_к$, можно для каждой функции ввести собственное оборудование. В таком процессоре любая команда последовательно проходит через все устройства, находясь на каждом этапе время Δt.

Так, команда с номером i поступает в УВК, через время Δt она переходит в УВА, а в УВК поступает команда с номером $i + 1$, затем через время Δt команда i поступает в УВО, $i + 1$ – в УВА, $i + 2$ – в УВК и т. д. Наконец, команда i поступает в АЛУ и через время Δt вырабатывается результат. После этого через время Δt будет получен результат команды $i + 1$. Таким образом, несмотря на то, что общее время выполнения любой команды сохранилось, результаты вырабатываются через время $\Delta t = t_к/n$, где n – число этапов этого конвейера команд.

Описанный принцип построения процессора, действительно, напоминает конвейер сборочного завода, на котором изделие последовательно проходит ряд рабочих мест. На каждом из этих мест над изделием производится новая операция. Эффект ускорения достигается за счет одновременной обработки ряда изделий на разных рабочих местах.

Конвейерные процессоры применяются во всех без исключения старших моделях семейств ЭВМ.

Временна́я диаграмма на рис. 1.11 строилась при следующих упрощениях: в потоке выбираемых из ПК команд отсутствуют команды условных переходов; все команды имеют одинаковое время нахождения на разных этапах.

Наличие команд условного перехода будет вынуждать переход к командам, которые в данный момент отсутствуют в конвейере, что потребует опустошения и повторного заполнения конвейера из ПК, а неодинаковая длина команд приведет к приостановкам конвейера. Такой в общем случае асинхронный характер функционирования конвейера снижает указанные выше цифры быстродействия.

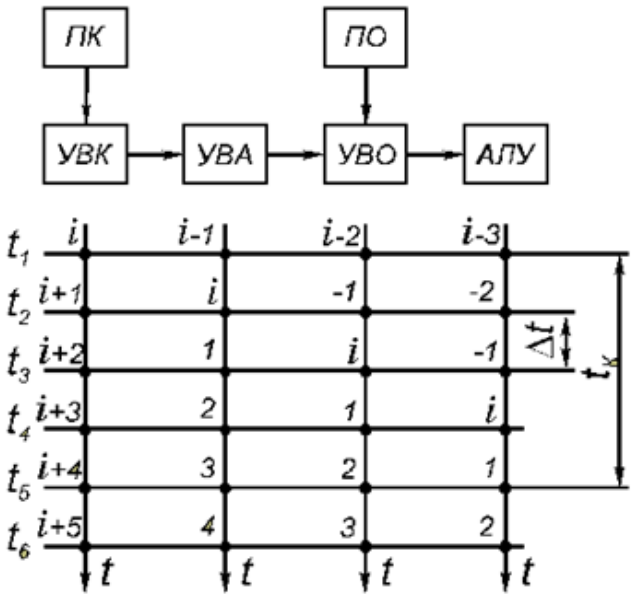


Рис. 1.11. Схема и функционирование конвейера команд:
ПК – память команд; ПО – память операндов; УВК, УВА, УВО – устройства выборки команд, вычисления адреса, выборки операндов соответственно

Стандартный способ увеличения быстродействия конвейерного процессора состоит в следующем: в существующем варианте конвейера выбирается устройство с наибольшим временем срабатывания и разделяется на два или более устройств с меньшим временем срабатывания каждое. При этом цикл конвейера Δt уменьшается. Если и после этого быстродействие конвейера недостаточно, снова выбирается наиболее медленное устройство и процесс повторяется.

Далее рассмотрим конвейеризацию устройств процессора в таком порядке: АЛУ, УВК, УВА, УВО.

Конвейеризация АЛУ

Арифметический конвейер можно построить для любых арифметико-логических операций – сложения, умножения, логических операций. В частности, на рис. 1.12, *а* показан конвейер для выполнения операции сложения двух чисел с плавающей запятой. Каждое число представлено в форме A*Rp, где A – мантисса; R – основание системы счисления, p – порядок. Конвейер для умножения целых чисел изображен на рис. 1.12, *б*. Здесь каждым входом сумматора Σ первого каскада управляет один разряд множителя. В зависимости от его значения на вход сумматора Σ подаются два смежных сдвинутых частичных произведения. Число каскадов такого конвейерного умножителя равно $\log_2 r$, где r – разрядность чисел Ai, Bi.

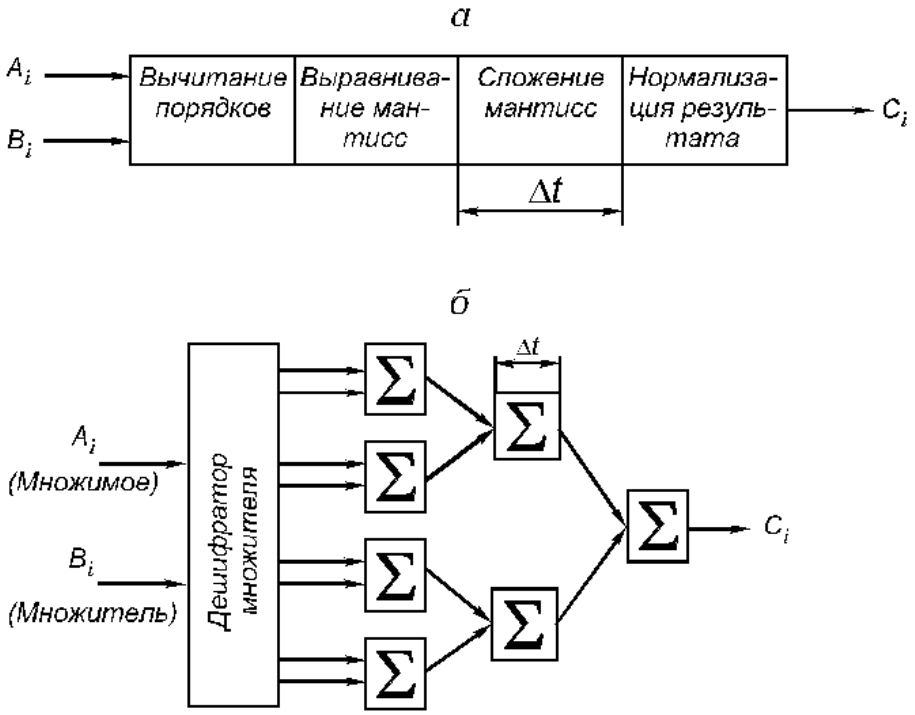


Рис. 1.12. Схема и функционирование арифметического конвейера для сложения (*а*) и умножения (*б*)

Как и в случае конвейера команд, числа поступают на вход конвейеризованного АЛУ вплотную друг за другом, поэтому результаты на выходе получаются с интервалом Δt. Для современных ЭВМ величина Δt стала меньше 10 нс, что соответствует быстродействию больше 100 млн оп/с, и цикл конвейера имеет тенденцию к дальнейшему уменьшению. Но при этом Δt не может стать меньше времени передачи данных с каскада на каскад.

Впервые арифметические конвейеры были использованы для целей обработки числовых векторов в ЭВМ STAR-100, запущенной в США в 1973 г.

Если ставится задача построить быстродействующий конвейерный процессор с V = 100 млн оп/с, то цикл всех его устройств не должен превышать Δt = 10 нс. Выше было показано, как обеспечить такой цикл в АЛУ.

Конвейеризация устройства выборки команд

Поскольку на каждый полученный в АЛУ результат приходится одна выборка команды из ПК, то время выборки этой команды не должно превышать интервала между получаемыми из АЛУ результатами. Однако по-

лупроводниковые запоминающие устройства большой емкости имеют цикл обращения ($t_{пк}$) во много раз больше требуемого цикла конвейера. Выходом здесь является использование множества автономных по функционированию блоков памяти. Число этих блоков $N = t_{пк} / \Delta t$ и может достигать величины 8...64 (обычно кратно степени 2).

Организация работы такой многоблочной памяти может быть различной. Если память имеет организацию, предназначенную для чтения со сдвигом (рис. 1.13), то в регистры адреса (РА) блоков памяти 1...4 с интервалом Δt подается новый адрес из счетчика адресов команд (СчАК) ПК. С таким же сдвигом по времени на выходе ПК будут появляться команды, которые затем поступают в буфер команд (БК), представляющий собой совокупность быстрых регистров. При поступлении каждой новой команды на вход БК содержимое всех его регистров сдвигается вверх на одну позицию и верхняя команда (самая старая) удаляется из БК.

В УВК имеется СчАК БК, который указывает положение в БК считываемой из УВА команды. При считывании из БК каждой команды его содержимое уменьшается на единицу, при добавлении в БК новой команды из ПК – увеличивается на единицу. За запросом новых команд в БК постоянно следит УВК, которое определяется величиной l. Если l становится меньше заданного уровня, то запускается СчАК ПК и производится выборка из ПК новых команд.

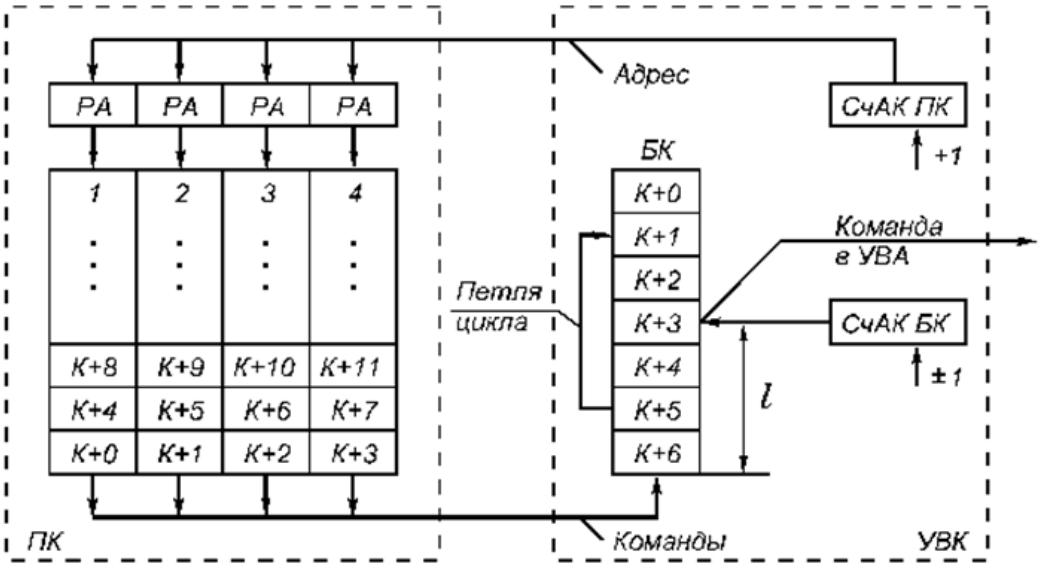


Рис. 1.13. Организация многоблочной памяти для выборки команд со сдвигом во времени

Во многих задачах линейной алгебры и задачах решения систем уравнений в частных производных общее время выполнения программ определяется скоростью выполнения внутренних циклов, число повторений

которых для задач большой размерности велико. Число же команд в петле цикла обычно невелико, и они полностью оказываются в БК. В таком случае выборка команд осуществляется только из БК, а ПК не используется. Это важно в структурах процессоров, где в качестве ПК и ПО используются одни и те же блоки памяти. В подобном случае выборка команд не будет создавать помех выборке операндов.

В схеме на рис. 1.14 за один цикл памяти в БК заносится несколько команд («широкое слово»), операции же в БК выполняются, как и ранее.

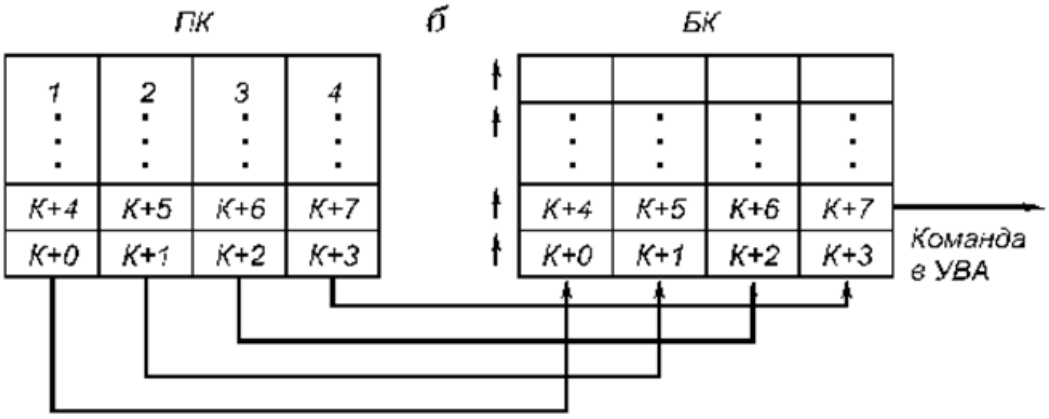


Рис. 1.14. Организация многоблочной памяти для выборки команд широким словом

Многоблочная структура памяти была впервые применена в ЭВМ STRETCH.

Конвейеризация устройства выборки адреса

Относительно УВА следует отметить, что оно является простым АЛУ для сложения коротких целых чисел (адресов), поэтому получение малого Δt для этого устройства не составляет труда.

Конвейеризация устройства выборки операндов

Сократить цикл работы УВО значительно сложнее. Здесь для уменьшения времени чтения операнда необходимо использовать многоблочную память. Однако между выборкой команд и выборкой операндов существует следующее принципиальное различие. Команды в программе и памяти располагаются в порядке линейного нарастания их номеров, поэтому во время исполнения текущей команды всегда можно вычислить адреса и выбрать (исключая команды переходов) любые следующие команды, что и приводит к уменьшению Δt при выборке команд. Однако строго упорядоченная выборка команд порождает неупорядоченную последовательность адресов для выборки операндов (рис. 1.15).

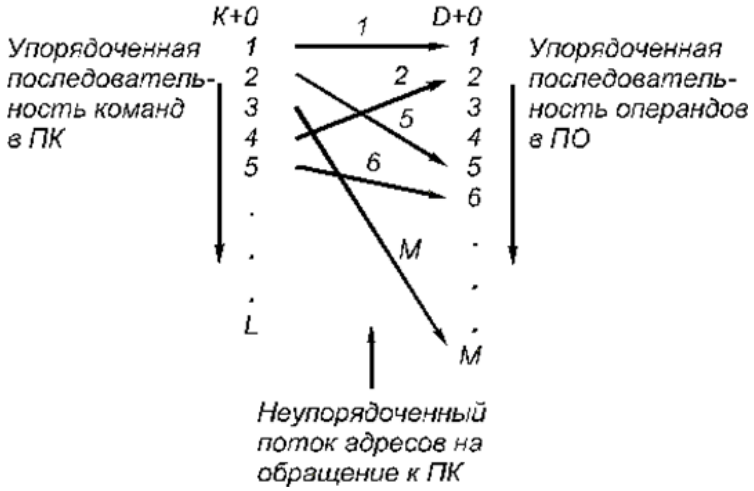


Рис. 1.15. Процесс генерации адресов операндов

Это означает, что выборка операндов для некоторой команды не может быть произведена заранее, до ее выборки. Следовательно, выборка операндов не может быть конвейеризована, поэтому для построения ПО используется не конвейерный, а поточный принцип организации многоблочной памяти (рис. 1.16).

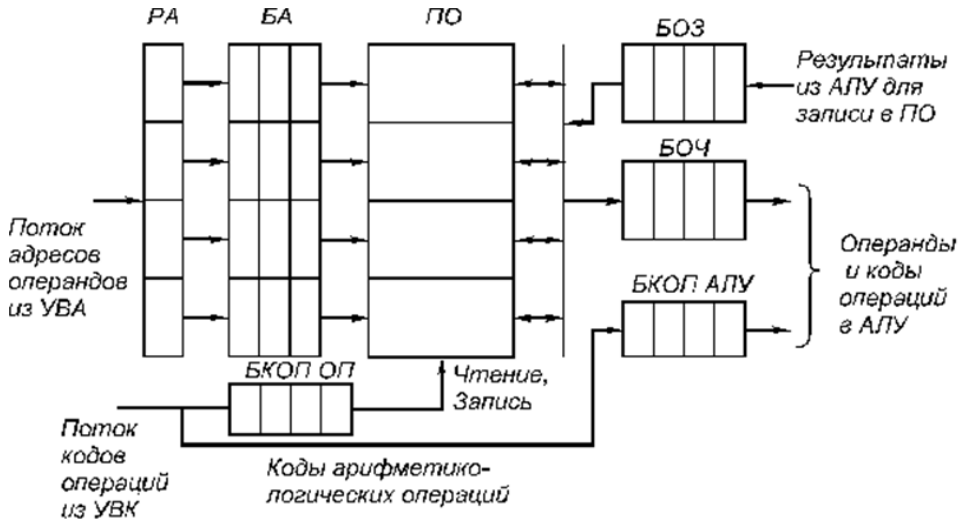


Рис. 1.16. Поточная организация устройства выборки операндов

Поступающие из УВА адреса операндов распределяются по блокам ПО. Поскольку распределение адресов носит достаточно случайный характер, в отдельных блоках памяти возможны очереди, для размещения которых введены буфера адресов (БА). Выбираемые из памяти операнды должны сразу поступать в АЛУ, однако вследствие неравномерности их появления из ПО и разной длительности исполнения операций в АЛУ вводятся буферные регистры операндов чтения (БОЧ) и записи (БОЗ). С каж-

дой парой операндов связан свой код операции, который хранится также в буфере кода операций (БКОП). Таким образом, в буфере операндов (БО) и БКОП хранятся готовые к исполнению в АЛУ группы информации.

Качественная картина сокращения цикла выборки из многоблочной памяти одного операнда представлена на рис 1.17. Время цикла одного блока памяти *t* выражено в относительных единицах.

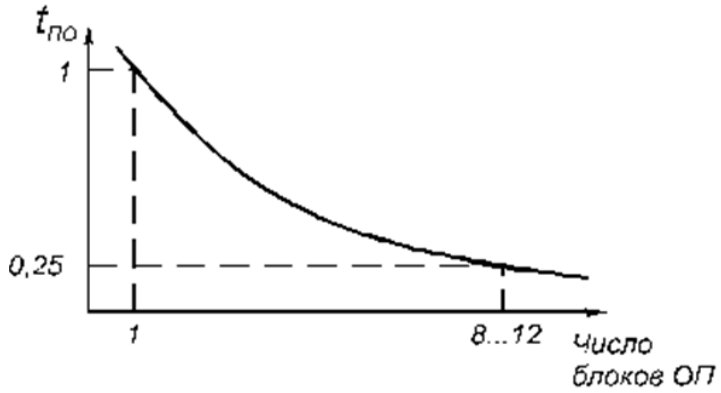


Рис. 1.17. Зависимость среднего времени выборки операнда из ПО от числа блоков ПО

1.5. Матричные вычислительные системы

В основе матричных систем лежит матричный процессор (*array processor*), состоящий из регулярного массива процессорных элементов (ПЭ). Организация систем подобного типа на первый взгляд достаточно проста. Они имеют общее управляющее устройство, генерирующее поток команд, и большое число ПЭ, работающих параллельно и обрабатывающих каждый свой поток данных. Однако на практике, чтобы обеспечить достаточную эффективность системы при решении широкого круга задач, необходимо организовать связи между процессорными элементами так, чтобы наиболее полно загрузить процессоры работой. Именно характер связей между ПЭ и определяет разные свойства системы.

Матричный процессор интегрирует множество идентичных функциональных блоков (ФБ), логически объединенных в матрицу и работающих в SIMD-стиле. Не столь существенно, как конструктивно реализована матрица процессорных элементов – на едином кристалле или на нескольких. Важен сам принцип – ФБ логически скомпонованы в матрицу и работают синхронно, т. е. присутствует только один поток команд для всех.

Структуру матричной вычислительной системы можно представить в виде, показанном на рис. 1.18.

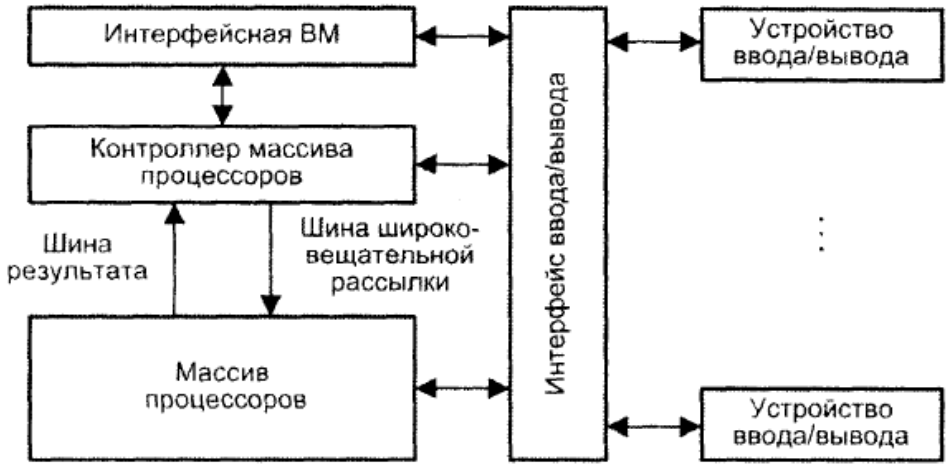


Рис. 1.18. Обобщенная модель матричной системы

Собственно параллельная обработка множественных элементов данных осуществляется массивом процессоров (МПр). Единый поток команд, управляющий обработкой данных в массиве процессоров, генерируется контроллером массива процессоров (КМП). КМП выполняет последовательный программный код, реализует операции условного и безусловного переходов, транслирует в МПр команды, данные и сигналы управления. Команды обрабатываются процессорами в режиме жесткой синхронизации. Сигналы управления используются для синхронизации команд и пересылок, а также для управления процессом вычислений, в частности, определяют, какие процессоры массива должны выполнять операцию, а какие – нет. Команды, данные и сигналы управления передаются из КМП в массив процессоров по шине широковещательной рассылки. Поскольку выполнение операций условного перехода зависит от результатов вычислений, результаты обработки данных в массиве процессоров транслируются в КМП, проходя по шине результата.

Для обеспечения пользователя удобным интерфейсом при создании и отладке программ в состав подобных ВС обычно включают интерфейсную ВМ (*front-end computer*). В роли такой ВМ выступает универсальная вычислительная машина, на которую дополнительно возлагается задача загрузки программ и данных в КМП. Кроме того, загрузка программ и данных в КМП может производиться и напрямую с устройств ввода/вывода, например, с магнитных дисков. После загрузки КМП приступает к выполнению программы, транслируя в МПр по широковещательной шине соответствующие команды.

Рассматривая массив процессоров, следует учитывать, что для хранения множественных наборов данных в нем, помимо множества процессоров, должно присутствовать и множество модулей памяти. Кроме того, в

массиве должна быть реализована сеть взаимосвязей, как между процессорами, так и между процессорами и модулями памяти. Таким образом, под термином «массив процессоров» понимают блок, состоящий из процессоров, модулей памяти и сети соединений.

Дополнительную гибкость при работе с рассматриваемой системой обеспечивает механизм маскирования, позволяющий привлекать к участию в операциях лишь определенное подмножество из входящих в массив процессоров. Маскирование реализуется как на стадии компиляции, так и на этапе выполнения, при этом процессоры, исключенные путем установки в ноль соответствующих битов маски, во время выполнения команды простаивают.

Интерфейсная ВМ

Интерфейсная ВМ (ИВМ) соединяет матричную систему с внешним миром, используя для этого какой-либо из сетевых интерфейсов, например Ethernet, как это имеет место в системе MasPar MP-1. Интерфейсная ВМ работает под управлением операционной системы, чаще всего ОС UNIX. На ИВМ пользователи подготавливают, компилируют и отлаживают свои программы. В процессе выполнения программы сначала загружаются из интерфейсной ВМ в контроллер управления массивом процессоров, который выполняет программу и распределяет команды и данные по процессорным элементам массива. В некоторых ВС, например в Massively Parallel Computer MPP, при создании, компиляции и отладке программ КМП и интерфейсная ВМ используются совместно.

На роль ИВМ подходят различные вычислительные машины. Так, в системе СМ-2 в этом качестве выступает рабочая станция SUN-4, в системе MasPar – ECstation 3000, а в системе MPP – DEC VAX-11/780.

Контроллер массива процессоров

Контроллер массива процессоров выполняет последовательный программный код, реализует команды ветвления программы, транслирует команды и сигналы управления в процессорные элементы. Рис. 1.19 иллюстрирует одну из возможных реализаций КМП, в частности, принятую в устройстве управления системы PASM.

При загрузке из ИВМ программа через интерфейс ввода/вывода заносится в оперативное запоминающее устройство КМП (ОЗУ КМП). Команды для процессорных элементов и глобальная маска, формируемая на этапе компиляции, также через интерфейс ввода/вывода загружаются в ОЗУ команд и глобальной маски (ОЗУ КГМ).

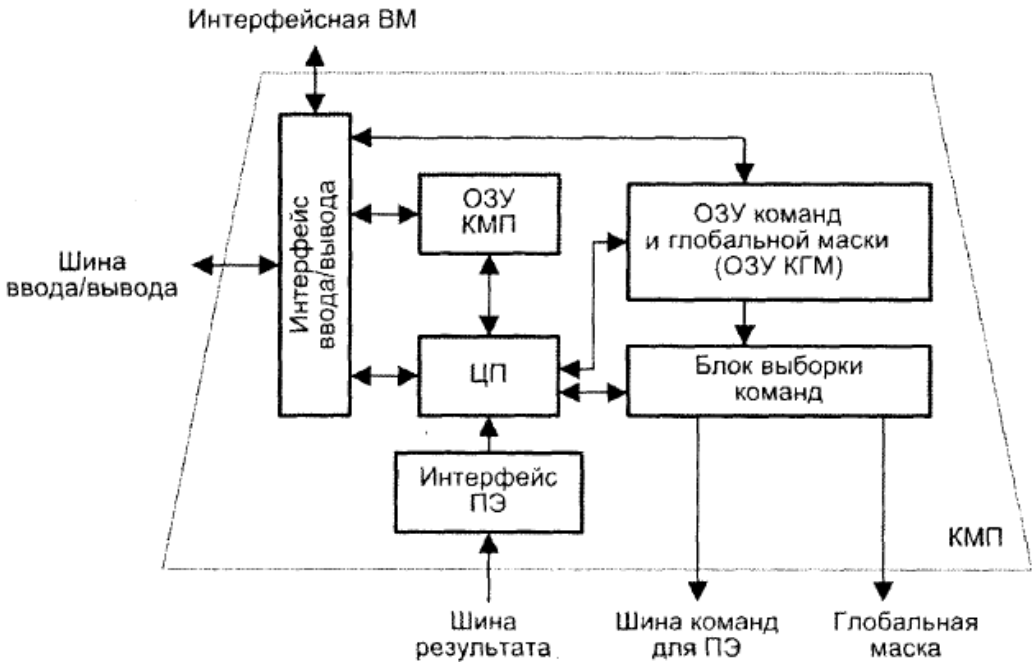


Рис. 1.19. Модель контроллера массива процессоров

Затем КМП начинает выполнять программу, извлекая либо одну скалярную команду из ОЗУ КМП, либо множественные команды из ОЗУ КГМ. Скалярные команды – команды, осуществляющие операции над хранящимися в КМП скалярными данными, выполняются центральным процессором (ЦП) контроллера массива процессоров. В свою очередь, команды, оперирующие параллельными переменными, хранящимися в каждом ПЭ, преобразуются в блоке выборки команд в более простые единицы выполнения – нанокоманды. Нанокоманды совместно с маской пересылаются через шину команд для ПЭ на исполнение в массив процессоров. Например, команда сложения 32-разрядных слов в КМП системы MPP преобразуется в 32 нанокоманды одноразрядного сложения, которые каждым ПЭ обрабатываются последовательно.

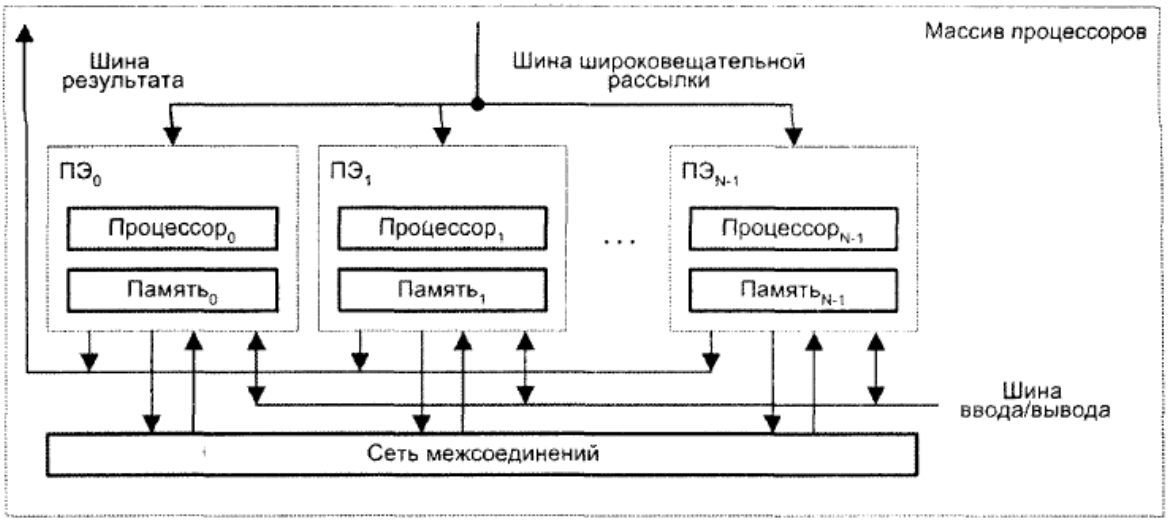
В большинстве алгоритмов дальнейший порядок вычислений зависит от результатов и/или флагов условий предшествующих операций. Для обеспечения такого режима в матричных системах статусная информация, хранящаяся в процессорных элементах, должна быть собрана в единое слово и передана в КМП для выработки решения о ветвлении программы. Например, в предложении IF A (условие A) THEN DO B оператор B будет выполнен, если условие A справедливо во всех ПЭ. Для корректного включения/отключения процессорных элементов КМП должен знать результат проверки условия A во всех ПЭ. Такая информация передается в КМП по однонаправленной шине результата. В системе CM-2 эта шина названа GLOBAL. В системе MPP для той же цели организована структура,

называемая деревом SUM-OR. Каждый ПЭ помещает содержимое своего одноразрядного регистра признака на входы дерева, которое с помощью операции логического сложения комбинирует эту информацию и формирует слово результата, используемое в КМП для принятия решения.

Массив процессоров

В матричных SIMD-системах распространение получили два основных типа архитектурной организации массива процессорных элементов (рис. 1.20).

В первом варианте, известном как архитектура типа «процессорный элемент – процессорный элемент» («ПЭ-ПЭ»), N процессорных элементов (ПЭ) связаны между собой сетью соединений (см. рис. 1.20, *а*).



а

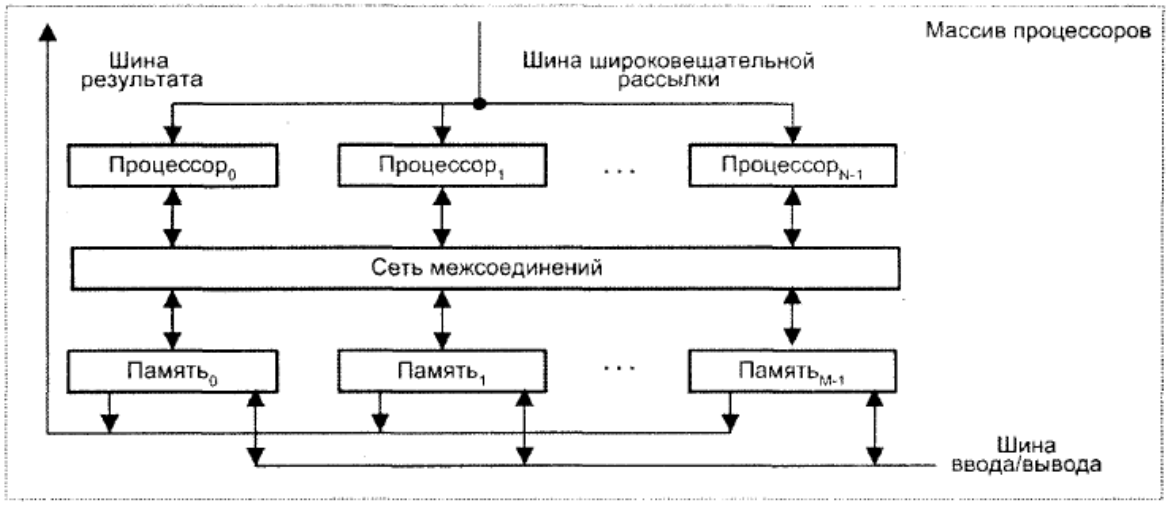


Рис. 1.20. Модели массивов процессоров: *а* – «процессорный элемент – процессорный элемент»; *б* – «процессор – память»

Каждый ПЭ – это процессор с локальной памятью. Процессорные элементы выполняют команды, получаемые из КМП по шине широковещательной рассылки, и обрабатывают данные, как хранящиеся в их локальной памяти, так и поступающие из КМП. Обмен данными между процессорными элементами производится по сети соединений, в то время как шина ввода/вывода служит для обмена информацией между ПЭ и устройствами ввода/вывода. Для трансляции результатов из отдельных ПЭ в контроллер массива процессоров служит шина результата. Благодаря использованию локальной памяти аппаратные средства ВС рассматриваемого типа могут быть построены весьма эффективно. Во многих алгоритмах действия по пересылке информации по большей части локальны, т. е. происходят между ближайшими соседями. По этой причине архитектура, где каждый ПЭ связан только с соседними, очень популярна.

В качестве примеров вычислительных систем с рассматриваемой архитектурой можно упомянуть MasPar MP-1, Connection Machine CM-2, GF11, DAP, MPP, STARAN, PEPE, ILLIAC IV.

Второй вид архитектуры – «процессор – память» – показан на рис. 1.20, *б*. В такой конфигурации двунаправленная сеть соединений связывает N процессоров с M модулями памяти. Процессоры управляются КМП через широковещательную шину. Обмен данными между процессорами осуществляется как через сеть, так и через модули памяти. Пересылка данных между модулями памяти и устройствами ввода/вывода обеспечивается шиной ввода/вывода. Для передачи данных из конкретного модуля памяти в КМП служит шина результата. Примерами ВС с рассмотренной архитектурой могут служить Burroughs Scientific Processor (BSP), Texas Reconfigurable Array Computer TRAC.

Структура процессорного элемента

В большинстве матричных SIMD-систем в качестве процессорных элементов применяются простые RISC-процессоры с локальной памятью ограниченной емкости. Например, каждый ПЭ системы MasPar MP-1 состоит из четырехразрядного процессора с памятью емкостью 64 Кбайт. В системе MPP используются одноразрядные процессоры с памятью 1 кбит каждый, а в CM-2 процессорный элемент представляет собой одноразрядный процессор с 64 Кбит локальной памяти. Благодаря простоте ПЭ массив может быть реализован в виде одной сверхбольшой интегральной микросхемы (СБИС). Это позволяет сократить число связей между микросхемами и, следовательно, габариты ВС. Так, одна СБИС в системе CM-2 содержит 16 процессоров (без блоков памяти), а в системе MasPar MP-1

СБИС состоит из 32 процессоров (также без блоков памяти). В системе MP-2 просматривается тенденция к применению более сложных микросхем, в частности, 32-разрядных процессоров с 256 Кбайт памяти в каждом.

Неотъемлемыми компонентами ПЭ (рис. 1.21) в большинстве вычислительных систем являются:

- арифметико-логическое устройство (АЛУ);
- регистры данных;
- сетевой интерфейс (СИ), который может включать в свой состав регистры пересылки данных;
- номер процессора;
- регистр флага разрешения маскирования (F);
- локальная память.

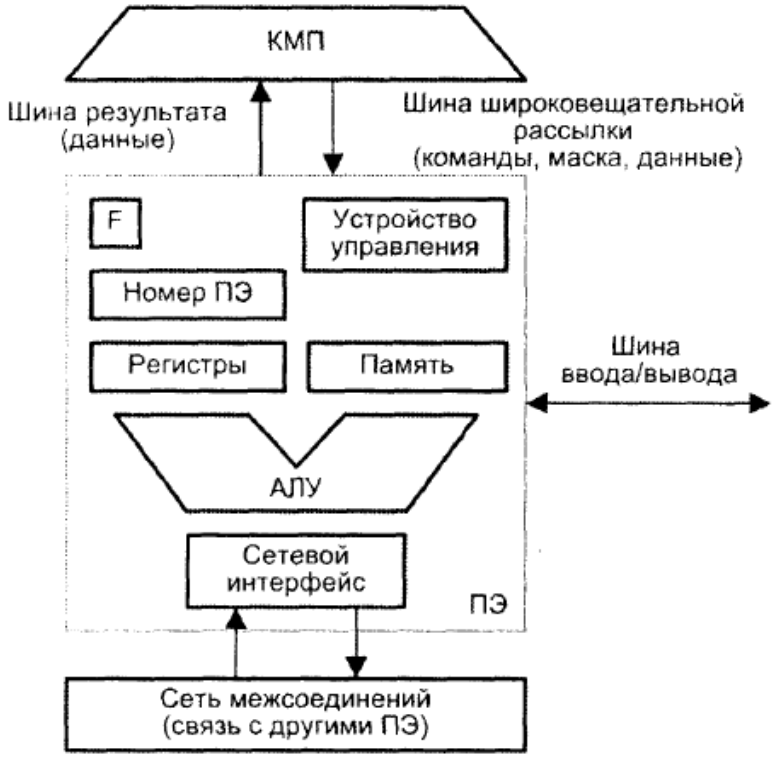


Рис. 1.21. Модель процессорного элемента

Процессорные элементы, управляемые командами, поступающими по широковещательной шине из КМП, могут выбирать данные из своей локальной памяти и регистров, обрабатывать их в АЛУ и сохранять результаты в регистрах и локальной памяти. ПЭ могут также обрабатывать те данные, которые поступают по шине широковещательной рассылки из КМП. Кроме того, каждый процессорный элемент вправе получать данные

из других ПЭ и отправлять их в другие ПЭ по сети соединений, используя для этого свой сетевой интерфейс. В некоторых матричных системах, в частности в MasPar MP-1, элемент данных из ПЭ-источника можно передавать в ПЭ-приемник непосредственно, в то время как в других, например в MPP, данные предварительно должны быть помещены в специальный регистр пересылки данных, входящий в состав сетевого интерфейса. Пересылка данных между ПЭ и устройствами ввода/вывода осуществляется через шину ввода/вывода BC. В ряде систем (MasPar MP-1) ПЭ подключены к шине ввода/вывода посредством сети соединений и канала ввода/вывода системы. Результаты вычислений любое ПЭ выдает в КМП через шину результата.

Каждому из N ПЭ в массиве процессоров присваивается уникальный номер, называемый также адресом ПЭ, который представляет собой целое число от 0 до N – 1. Чтобы указать, должен ли данный ПЭ участвовать в общей операции, в его составе имеется регистр флага разрешения F. Состояние этого регистра определяют сигналы управления из КМП либо результаты операций в самом ПЭ, либо и те, и другие совместно.

Еще одной существенной характеристикой матричной системы является способ синхронизации работы ПЭ. Так как все ПЭ получают и выполняют команды одновременно, их работа жестко синхронизируется. Это особенно важно в операциях пересылки информации между ПЭ. В системах, где обмен производится с четырьмя соседними ПЭ, передача информации осуществляется в режиме «регистр – регистр».

Подключение и отключение процессорных элементов

В процессе вычислений в ряде операций должны участвовать только определенные ПЭ, в то время как остальные ПЭ остаются бездействующими. Разрешение и запрет работы ПЭ могут исходить от контроллера массива процессоров (глобальное маскирование) и реализуются с помощью схем маскирования ПЭ. В этом случае решение о необходимости маскирования принимается на этапе компиляции кода. Решение о маскировании может также приниматься во время выполнения программы (маскирование, определяемое данными), при этом опираются на хранящийся в ПЭ флаг разрешения маскирования F.

При маскировании, определяемом данными, каждый ПЭ самостоятельно объявляет свой статус «подключен/не подключен». В составе системы команд имеются наборы маскируемых и немаскируемых команд. Маскируемые команды выполняются в зависимости от состояния флага F, в то время как немаскируемые флаг просто игнорируют. Процедуру мас-

кирования рассмотрим на примере предложения IF-THEN-ELSE. Пусть x – локальная переменная (хранящаяся в локальной памяти каждого ПЭ). Предположим, что процессорные элементы массива параллельно выполняют ветвление

IF (x > 0) THEN <оператор A> ELSE – <оператор B>

и каждый ПЭ оценивает условие IF. Те ПЭ, для которых условие $x > 0$ справедливо, установят свой флаг F в единицу, тогда как остальные ПЭ – в ноль. Далее КМП распределяет оператор A по всем ПЭ. Команды, реализующие этот оператор, должны быть маскируемыми. Оператор A будет выполнен только теми ПЭ, где флаг F установлен в единицу. Далее КМП передает во все ПЭ немаскируемую команду ELSE, которая заставит все ПЭ инвертировать состояние своего флага F. Затем КМП транслирует во все ПЭ оператор B, который также должен состоять из маскируемых команд. Оператор будет выполнен теми ПЭ, где флаг F после инвертирования был установлен в единицу, то есть где результат проверки условия $x > 0$ был отрицательным.

При использовании схемы глобального маскирования контроллер массива процессоров вместе с командами посылает во все ПЭ глобальную маску. Каждый ПЭ декодирует эту маску и по результату выясняет, должен ли он выполнять данную команду или нет.

В зависимости от способа кодирования маски существует несколько различных схем глобального маскирования. В схеме, примененной в вычислительной системе ILLIAC IV с 64-мя 64-разрядными ПЭ, маска представляет собой N-разрядный вектор. Каждый бит вектора отражает состояние одного ПЭ. Если бит содержит единицу, соответствующий ПЭ будет активным, в противном случае – пассивным. Несмотря на свою универсальность, при больших значениях N схема становится неудобной. В варианте маскирования с адресом ПЭ используется $2m$-разрядная маска (m = log2N), в которой каждая позиция соответствует одному разряду в двоичном представлении адреса ПЭ. Каждая позиция может содержать 0, i или X. Таким образом, маска состоит из $2m$ битов. Если для всех i ($0 \le i < m$) i-я позиция в маске и i-я позиция в адресе ПЭ совпадают или в i-й позиции маски стоит X, ПЭ будет активным. Например, маска 000X1 представляет процессорные элементы с номерами 1 и 3, в то время как маска XXXXO представляет все ПЭ с четными номерами (все это для массива из 32 ПЭ). Здесь можно активизировать только подмножество из всех возможных комбинаций процессорных элементов массива, что на практике не является ограничением, так как в реальных алгоритмах обычно участвуют не произвольные ПЭ, а лишь расположенные регулярным образом.

Глобальные и локальные схемы маскирования могут комбинироваться. В таком случае активность ПЭ в равной мере определяется как флагом F, так и глобальной маской.

Сети взаимосвязей процессорных элементов

Эффективность сетей взаимосвязей процессорных элементов во многом определяет возможную производительность всей матричной системы. Применение находят самые разнообразные топологии сетей.

Поскольку процессорные элементы в матричных системах функционируют синхронно, обмениваться информацией они также должны по согласованной схеме, причем необходимо обеспечить возможность синхронной передачи от нескольких ПЭ-источников к одному ПЭ-приемнику. Когда для передачи информации в сетевом интерфейсе задействуется только один регистр пересылки данных, это может привести к потере данных, поэтому в ряде ВС для предотвращения подобной ситуации предусмотрены специальные механизмы. Так, в системе СМ-2 используется оборудование, объединяющее сообщения, поступившие к одному ПЭ. Объединение реализуется за счет операций арифметического и логического сложения, наложения записей, нахождения меньшего и большего из двух значений. В некоторых системах, например МР-1, имеется возможность записать одновременно пришедшие сообщения в разные ячейки локальной памяти.

Хотя пересылки данных по сети инициируются только активными ПЭ, пассивные процессорные элементы также вносят вклад в эти операции. Если активный ПЭ инициирует чтение из другого ПЭ, операция выполняется вне зависимости от статуса ПЭ, из которого считывается информация. То же самое происходит и при записи.

Наиболее распространенными топологиями в матричных системах являются решетчатые и гиперкубические. Так, в ILLIAC IV, MPP и СМ-2 каждый ПЭ соединен с четырьмя соседними. В МР-1 и МР-2 каждый ПЭ связан с восемью смежными ПЭ. В ряде систем реализуются многоступенчатые динамические сети соединений (МР-1, МР-2, GF11).

1.6. Ассоциативные и систолические системы

Ассоциативные вычислительные системы

Ассоциативные системы, как и матричные, характеризуются большим числом операционных устройств, способных одновременно, по командам одного управляющего устройства, вести обработку нескольких потоков данных. Но эти системы существенно отличаются от матричных способами формирования потоков данных. В матричных системах данные

поступают на обработку от общих или раздельных запоминающих устройств с адресной выработкой информации либо непосредственно от устройств – источников данных. В ассоциативных системах информация на обработку поступает от ассоциативных запоминающих устройств (АЗУ), характеризующихся тем, что информация из них выбирается не по определенному адресу, а по ее содержанию.

Принцип работы АЗУ поясняет схема, представленная на рис. 1.22. Запоминающий массив, как и в адресных ЗУ, разделен на *m*-разрядные ячейки, число которых *n*. Практически для любого типа АЗУ характерно наличие следующих элементов: запоминающего массива; регистра ассоциативных признаков (РгАП); регистра маски (РгМ); регистра индикаторов адреса со схемами сравнения на входе. В АЗУ могут быть и другие элементы, наличие и функции которых определяются способом использования АЗУ.

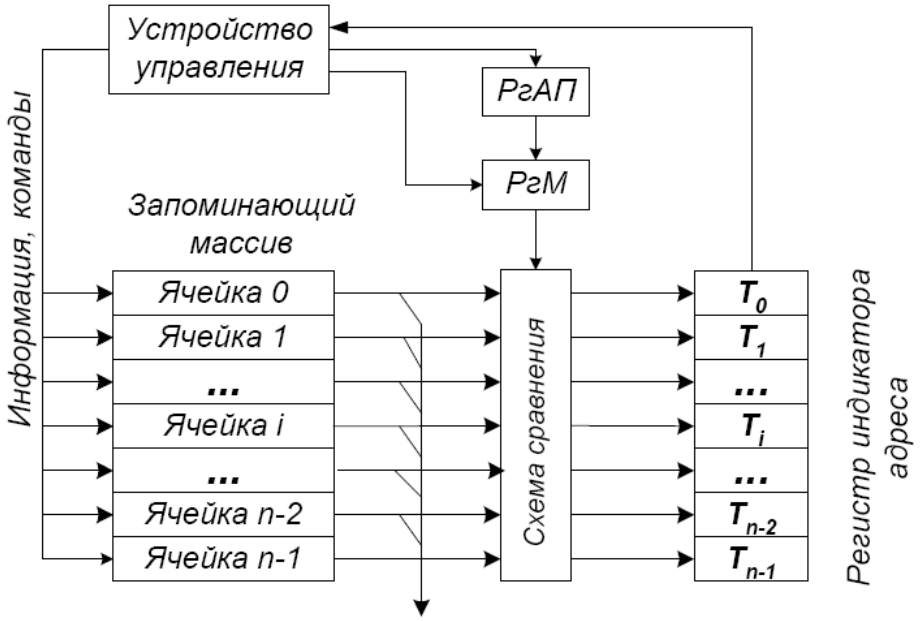


Рис. 1.22. Ассоциативное запоминающее устройство

Выборка информации из АЗУ происходит следующим образом. В РгАП из устройства управления передается код признака искомой информации (иногда его называют компарандом). Код может иметь произвольное число разрядов – от 1 до *m*. Если код признаков используется полностью, то он без изменения поступает на схему сравнения, если же необходимо использовать только часть кода, тогда ненужные разряды маскируются с помощью РгМ. Перед началом поиска информации в АЗУ все разряды регистра индикаторов адреса устанавливаются в состояние 1. После

этого производится опрос первого разряда всех ячеек ЗМ и содержимое сравнивается с первым разрядом РгАП. Если содержимое первого разряда *i*-й ячейки не совпадает с содержимым первого разряда РгАП, то соответствующий этой ячейке разряд регистра индикаторов адреса Ti сбрасывается в состояние 0, если совпадает, на Ti остается 1. Затем эта операция повторяется со вторым, третьим и последующими разрядами до тех пор, пока не будет произведено сравнение со всеми разрядами РгАП. После поразрядного опроса и сравнения в состоянии 1 останутся те разряды регистра индикаторов адреса, которые соответствуют ячейкам, содержащим информацию, совпадающую с записанной в РгАП. Эта информация может быть считана в той последовательности, которая определяется устройством управления.

Заметим, что время поиска информации в ЗМ по ассоциативному признаку зависит только от числа разрядов признака и от скорости опроса разрядов, но совершенно не зависит от числа ячеек ЗМ. Этим и определяется главное преимущество АЗУ перед адресными ЗУ: в адресных ЗУ при операции поиска необходим перебор всех ячеек запоминающего массива.

Запись новой информации в ЗМ производится без указания номера ячейки. Обычно один из разрядов каждой ячейки используется для указания ее занятости, т. е. если ячейка свободна для записи, то в этом разряде записан 0, а если занята – 1. Тогда при записи в АЗУ новой информации устанавливается признак 0 в соответствующем разряде РгАП и определяются все ячейки ЗМ, которые свободны для записи. В одну из них устройство управления помещает новую информацию.

Нередко АЗУ строятся таким образом, что кроме ассоциативной допускается и прямая адресация данных, что представляет определенные удобства при работе с периферийными устройствами.

Необходимо отметить, что запоминающие элементы АЗУ в отличие от элементов адресуемых ЗУ должны не только хранить информацию, но и выполнять определенные логические функции, поэтому позволяют осуществить поиск не только по равенству содержимого ячейки заданному признаку, но и по другим условиям: содержимое ячейки больше (меньше) признака РгАП, а также больше или равно (меньше или равно).

Отмеченные выше свойства АЗУ характеризуют преимущества АЗУ для обработки информации. Формирование нескольких потоков идентичной информации с помощью АЗУ осуществляется быстро и просто, а с большим числом операционных элементов можно создавать высокопроизводительные системы. Надо учитывать еще и то, что на основе ассоциативной памяти легко реализуется изменение места и порядка расположе-

ния информации. Благодаря этому АЗУ является эффективным средством формирования наборов данных.

Исследования показывают, что целый ряд задач, таких, как обработка радиолокационной информации, распознавание образов, обработка различных снимков и других задач с матричной структурой данных, эффективно решается ассоциативными системами. К тому же программирование таких задач для ассоциативных систем гораздо проще, чем для традиционных.

Наиболее характерным представителем группы ассоциативных вычислительных систем является система STARAN, разработанная в США. От матричных систем, описанных выше, она отличается не только наличием ассоциативной памяти, но и другими особенностями. Ассоциативная память является памятью с многомерным доступом, т. е. в нее можно обратиться как поразрядно, так и пословно, операционные процессорные элементы предусмотрены для каждого слова памяти; имеется уникальная схема перестановок для перегруппировки данных в памяти.

Основным элементом системы является многомерная ассоциативная матрица – ассоциативный модуль (АМ), который представляет собой квадрат из 256 разрядов на 256 слов, т. е. содержит в общей сложности 65536 бит данных. Для обработки информации имеется 256 процессорных элементов, которые последовательно, разряд за разрядом, обрабатывают слова (рис. 1.23). Все ПЭ работают одновременно, по одной команде, выдаваемой устройством управления. Таким образом, сразу по одной команде обрабатываются все выбранные по определенным признакам из памяти слова.

Схема перестановок позволяет сдвигать и перегруппировывать данные так, чтобы над словами, хранящимися в памяти, можно было выполнять параллельно арифметические и логические операции. Бóльшая часть операций выполняется в отношении каждого из 256-разрядных слов. Операции, в которых участвуют несколько слов, используются достаточно редко. Обычно 250-разрядное слово ассоциативной матрицы разбивается программистом на поля переменной длины и в процессе обработки именно над этими полями производятся и арифметические и логические действия.

Базовая конфигурация системы STARAN содержит один АМ. Однако число этих модулей может варьироваться в системе от 1 до 32. Таким образом, при максимальной комплектации в системе может подвергаться ассоциативной обработке 256 кбайт информации. Скорость поиска и обработки информации 256-ю процессорными элементами высока, и остальные элементы системы спроектированы так, чтобы поддерживать эту скорость.

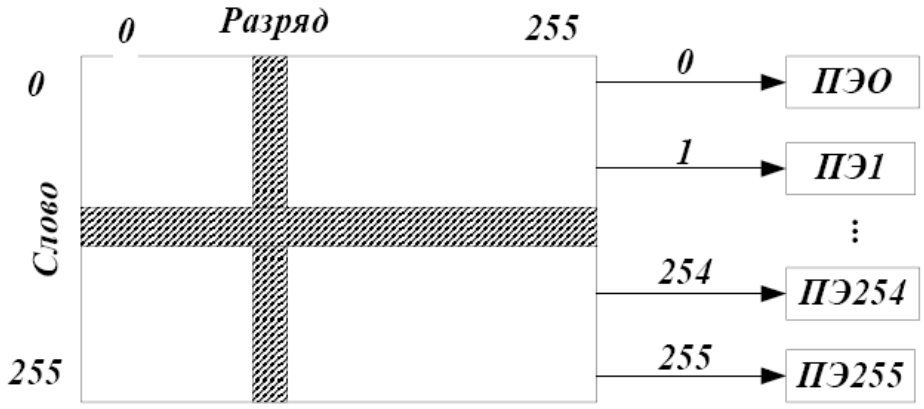


Рис. 1.23. Процессорная обработка в системе STARAN

Устройство управления ассоциативными модулями организует выполнение операций над данными по командам, хранящимся в управляющей памяти. Оно может выбирать несколько рабочих подмножеств из общего множества данных, хранимых в АМ, и выполнять над этими подсистемами операции, не затрагивая остальную информацию.

Управляющая память разделена на шесть секций: первая (емкостью 612 слов) – память библиотеки подпрограмм; вторая и третья (512 слов) – память команд; четвертая (512 слов) – быстродействующий буфер данных; пятая (16384 слов) – основная память; шестая (10720 слов) – область памяти для прямого доступа. Длина одного слова – 32 разряда. Первые четыре секции выполнены на интегральных схемах и имеют высокое быстродействие с длительностью цикла памяти около 200 нс. Вторая и третья секции (память команд) работают попеременно – одна выдает команды в УУ, а другая в это время загружается от страничного устройства, и наоборот. Пятая и шестая секции выполнены на ферритовых сердечниках, длительность цикла примерно 1 мкс. При необходимости емкость пятой секции может быть удвоена. Страничное устройство загружает первые три секции памяти информацией из быстродействующего буфера, основной памяти или памяти прямого доступа.

Последовательный контроллер ассоциативной системы является обычной однопроцессорной ЭВМ типа РДР-11 и обеспечивает работу в режиме трансляции и отладки программ; первоначальную загрузку управляющей памяти, связь между оператором и системой; управление программами обработки прерываний по ошибкам, а также программами технической диагностики обслуживания. Последовательный контроллер снабжен памятью (емкость 8 КСлов), печатающим устройством, перфоленточным вводом – выводом и имеет интерфейс, обеспечивающий связь с другими элементами системы.

Подсистема ввода – вывода обеспечивает возможность подключения к системе STARAN других вычислительных устройств и разнообразного периферийного оборудования. Имеются четыре вида интерфейсов: прямой доступ к памяти; буферизованный ввод – вывод; параллельный ввод – вывод; логическое устройство внешних функций. Прямой доступ к памяти позволяет использовать память внешней (несистемной) ЭВМ как часть управляющей памяти системы. Эта память становится таким образом доступной как для внешней ЭВМ, так и для системы SPARAN. При этом нет необходимости в буферизации передаваемой между ними информации.

Интерфейс прямого доступа может использоваться и для подключения внешней памяти. Буферизованный ввод – вывод используется для связи системы со стандартными периферийными устройствами, обмен производится блоками данных или программ. Этот интерфейс может использоваться и для связи с несистемной ЭВМ, однако прямой доступ там все-таки предпочтителен, так как обмен производится быстрее и нет необходимости формирования информации в блоки перед передачей. Параллельный ввод – вывод, который включает в себя по 256 входов и 256 выходов для каждой матрицы, является важной составной частью подсистемы ввода – вывода. Он позволяет увеличить скорость передачи данных между матрицами, обеспечить связь системы с высокоскоростными средствами ввода – вывода и непосредственную связь любого устройства с ассоциативными модулями. С помощью параллельного ввода – вывода можно, в частности, подключать к ассоциативным матрицам накопители на магнитных дисках, что позволяет быстро вводить и выводить большие объемы информации.

Совокупность всех перечисленных средств, входящих в систему STARAN, позволяет выполнять одновременно сотни и тысячи одинаковых операций при решении определенных классов задач.

Вычислительные системы с систолической структурой

В фон-неймановских машинах данные, считанные из памяти, однократно обрабатываются в процессорном элементе, после чего снова возвращаются в память (рис. 1.24, *а*). Авторы идеи систолической матрицы Кунг и Лейзерсон предложили организовать вычисления так, чтобы данные на своем пути от считывания из памяти до возвращения обратно пропускались через как можно большее число ПЭ (см. рис. 1.24, *б*).

Если сравнить положение памяти в ВС со структурой живого организма, то по аналогии ей можно отвести роль сердца, множеству ПЭ – роль тканей, а поток данных рассматривать как циркулирующую кровь. Отсюда и происходит название «систолическая матрица» (систола – сокращение предсердий и желудочков сердца, при котором кровь нагнетается в артерии).

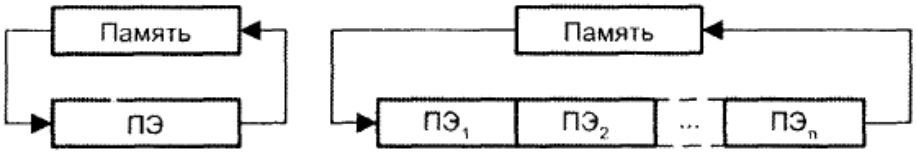


Рис. 1.24. Обработка данных в ВС: *а* – фон-неймановского типа; *б* – систолической структуры

Систолические структуры эффективны при выполнении матричных вычислений, обработке сигналов, сортировке данных и т. д. В качестве примера авторами идеи был предложен линейный массив для алгоритма матричного умножения, показанный на рис. 1.25.

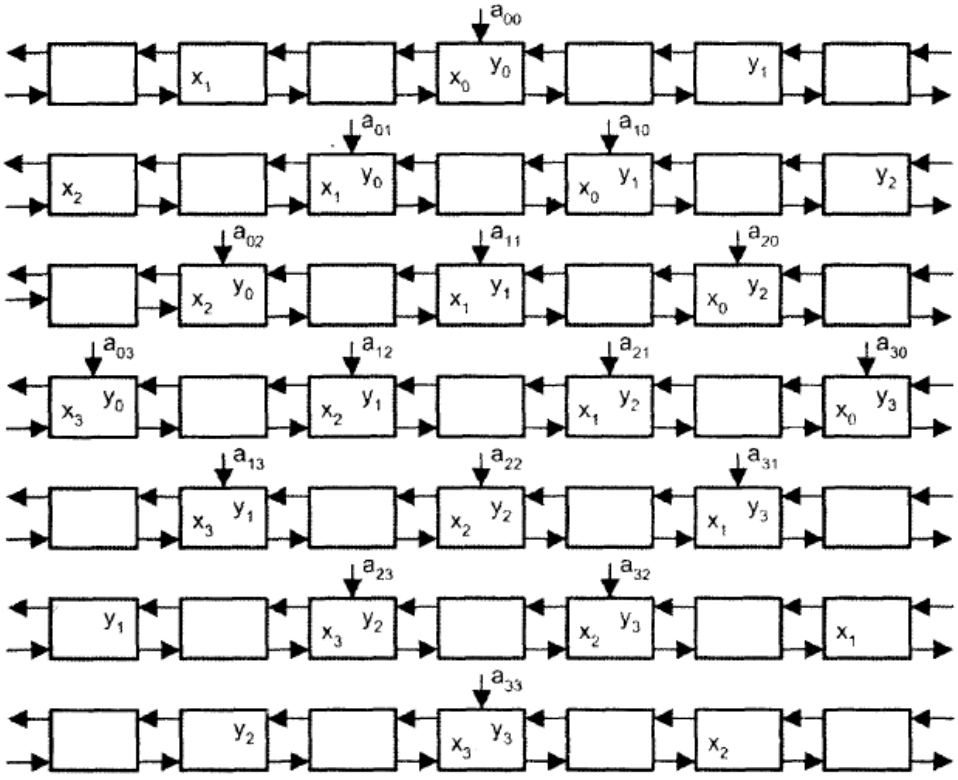


Рис. 1.25. Процесс векторного умножения матриц (n = 4)

В основе схемы лежит ритмическое прохождение двух потоков данных, x_i и y_i, навстречу друг другу. Последовательные элементы каждого потока разделены одним тактовым периодом, чтобы любой из них мог встретиться с любым элементом встречного потока. Если бы они следовали в каждом периоде, то элемент x_i никогда бы не встретился с элементами y: y_i + l, y_i + 3... Вычисления выполняются параллельно в процессорных элементах, каждый из которых реализует один шаг в операции вычисления скалярного произведения (IPS, Inner Product Step) и носит название IPS-элемента (рис. 1.26).

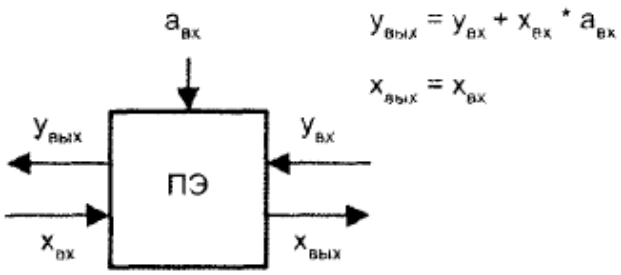


Рис. 1.26. Функциональная схема IPS-элемента

Значение $y_{вх}$, поступающее на вход ПЭ, суммируется с произведением входных значений $x_{вх}$ и $a_{вх}$. Результат выходит из ПЭ как $y_{вых}$. Значение $x_{вх}$, кроме того, для возможного последующего использования остальной частью массива транслируется через ПЭ без изменений и покидает его в виде $x_{вых}$.

Таким образом, систолическая структура – это однородная вычислительная среда из процессорных элементов, совмещающая в себе свойства конвейерной и матричной обработки и обладающая следующими особенностями:

- вычислительный процесс в систолических структурах представляет собой непрерывную и регулярную передачу данных от одного ПЭ к другому без запоминания промежуточных результатов вычисления;
- каждый элемент входных данных выбирается из памяти однократно и используется столько раз, сколько необходимо по алгоритму, ввод данных осуществляется в крайние ПЭ матрицы;
- образующие систолическую структуру ПЭ однотипны, и каждый из них может быть менее универсальным, чем процессоры обычных многопроцессорных систем;
- потоки данных и управляющих сигналов обладают регулярностью, что позволяет объединять ПЭ локальными связями минимальной длины;
- алгоритмы функционирования позволяют совместить параллелизм с конвейерной обработкой данных;
- производительность матрицы можно улучшить за счет добавления в нее определенного числа ПЭ, причем коэффициент повышения производительности при этом линеен.

В настоящее время достигнута производительность систолических процессоров порядка 1000 млрд операций/с.

Классификация систолических структур

Анализ различных типов систолических структур и тенденций их развития позволяет классифицировать эти структуры по нескольким признакам.

По степени гибкости систолические структуры могут быть подразделены:

- на специализированные;
- на алгоритмически ориентированные;
- на программируемые.

Специализированные структуры ориентированы на выполнение определенного алгоритма. Эта ориентация отражается не только в конкретной геометрии систолической структуры, статичности связей между ПЭ и числе ПЭ, но и в выборе типа операции, выполняемой всеми ПЭ. Примерами являются структуры, ориентированные на рекурсивную фильтрацию, быстрое преобразование Фурье для заданного количества точек, конкретные матричные преобразования.

Алгоритмически ориентированные структуры обладают возможностью программирования либо конфигурации связей в систолической матрице, либо самих ПЭ. Возможность программирования позволяет выполнять на таких структурах некоторое множество алгоритмов, сводимых к однотипным операциям над векторами, матрицами и другими числовыми множествами.

В программируемых систолических структурах имеется возможность программирования как самих ПЭ, так и конфигурации связей между ними. При этом ПЭ могут обладать локальной памятью программ, и хотя все они имеют одну и ту же организацию, в один и тот же момент времени допускается выполнение различных операций из некоторого набора. Команды или управляющие слова, хранящиеся в памяти программ таких ПЭ, могут изменять и направление передачи операндов.

По разрядности процессорных элементов систолические структуры делятся:

- на одноразрядные;
- на многоразрядные.

В одноразрядных матрицах ПЭ в каждый момент времени выполняет операцию над одним двоичным разрядом; а в многоразрядных – над словами фиксированной длины.

По характеру локально-пространственных связей систолические структуры бывают:

- одномерные;
- двухмерные;
- трехмерные.

Выбор структуры зависит от вида обрабатываемой информации. Одномерные схемы применяются при обработке векторов, двухмерные – матриц, трехмерные – множеств иного типа.

Топология систолических структур

В настоящее время разработаны систолические матрицы с различной геометрией связей – линейные, квадратные, гексагональные, трехмерные и др. Перечисленные конфигурации систолических матриц приведены на рис. 1.27.

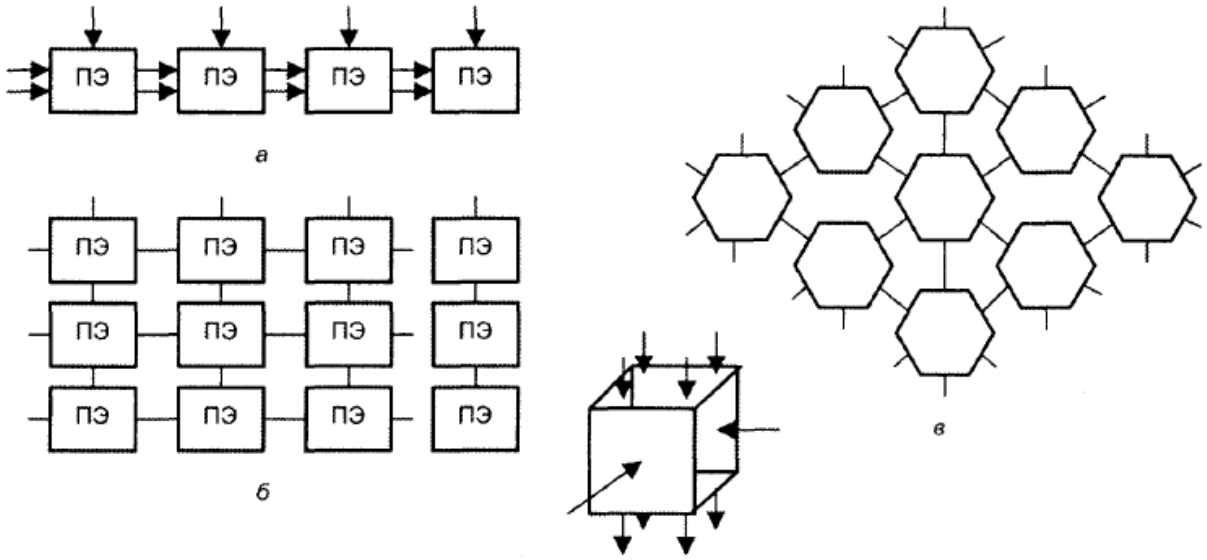


Рис. 1.27. Конфигурация систолических матриц: *а* – линейная; *б* – прямоугольная; *в* – гексагональная; *г* – трехмерная

Каждая конфигурация матрицы наиболее приспособлена для выполнения определенных функций, например, линейная матрица оптимальна для реализации фильтров в реальном масштабе времени; гексагональная – для выполнения операций обращения матриц, а также действий над матрицами специального вида (Теплица – Генкеля); трехмерная – для нахождения значений нелинейных дифференциальных уравнений в частных производных или для обработки сигналов антенной решетки. Наиболее универсальными и наиболее распространенными, тем не менее, можно считать матрицы с линейной структурой.

Для решения сложных задач конфигурация систолической структуры может представлять собой набор отдельных матриц, сложную сеть взаимосвязанных матриц либо обрабатывающую поверхность. Под обрабатывающей поверхностью понимается бесконечная прямоугольная сетка ПЭ, где каждый ПЭ соединяется со своими четырьмя соседями (или бóльшим

числом ПЭ). Одним из наиболее подходящих элементов для реализации обрабатывающей поверхности является матрица простых ПЭ или транспьютеров.

Учитывая то, что матрицы ПЭ обычно реализуются на основе сверхбольших интегральных схем, возникающие при этом ограничения привели к тому, что наиболее распространены матрицы с одним, двумя и тремя трактами данных и с одинаковым либо противоположным направлением передачи, обозначаемые как ULA, BLA и TLA соответственно.

ULA (*Unidirectional Linear Array*) – это однонаправленный линейный процессорный массив, где потоки данных перемещаются в одном направлении. ПЭ в массиве могут быть связаны одним, двумя или тремя трактами.

При реализации алгоритма векторного произведения матриц один из потоков данных перемещается вправо, в то время как второй резидентно расположен в массиве (рис. 1.28). Используемый ПЭ представляет собой модифицированный IPS-элемент, поскольку имеется только один тракт данных, а элементы второго потока хранятся в ПЭ массива.

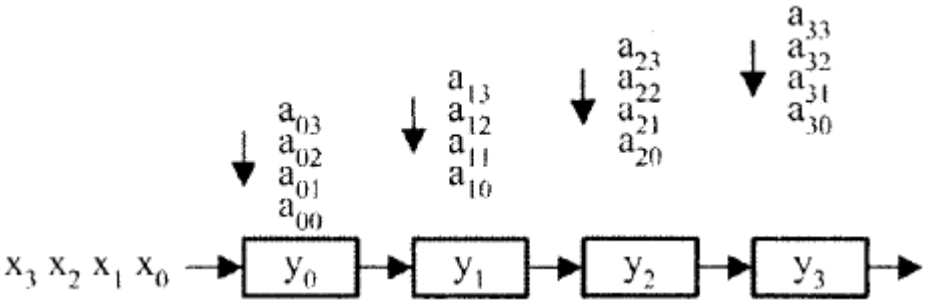


Рис. 1.28. Поток данных при векторном перемножении матриц ULA (n = 4)

BLA (*Bidirectional Linear Array*) – это двунаправленный линейный процессорный массив, в котором два потока данных движутся навстречу друг другу. BLA, где один из потоков является выходным, называется *регулярным*.

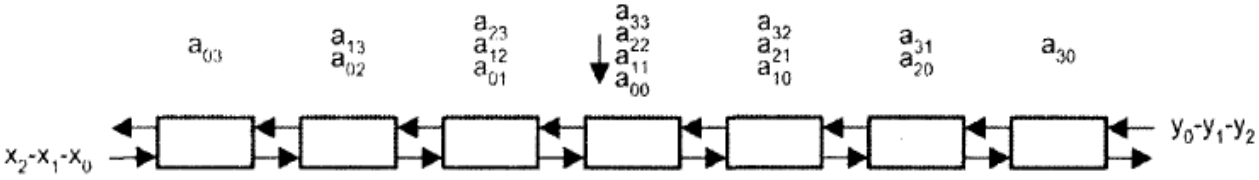


Рис. 1.29. Поток данных при векторном перемножении матриц в BLA (n = 4)

Реализация рассмотренной ранее операции с применением BLA показана на рис. 1.29. В версии ULA процессоры используются более эффек-

тивно, поскольку в них элементы потока следуют в каждом такте, а не через такт, как в BLA.

TLA (*Three-path communication Linear Array*) – линейный процессорный массив с тремя коммуникационными трактами, в котором по разным направлениям перемещаются три потока данных. На рис. 1.30 показан пример фильтра ARMA, предложенного Кунгом и построенного по схеме TLA. Возможны несколько вариантов такого фильтра, в зависимости от числа выходных потоков данных и от значений, хранящихся в памяти (в примере фигурирует один выходной поток).

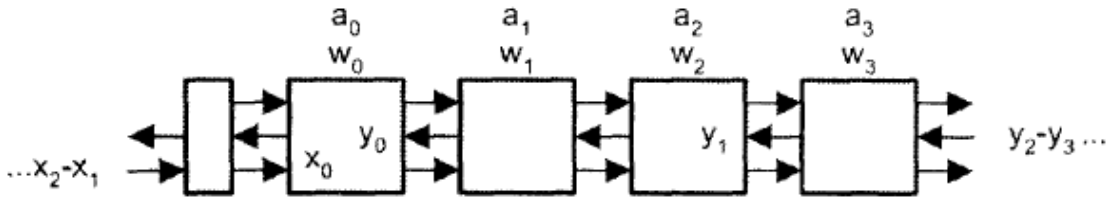


Рис. 1.30. Поток данных в TLA фильтра ARMA (n = 3)

Процессорные элементы выполняют две операции IPS и обычно называются сдвоенными IPS-элементами. Две версии таких ПЭ представлены на рис. 1.31. ПЭ могут использовать как хранимые в памяти значения (см. рис. 1.31, *а*, *б*), так и внешние данные (см. рис. 13.21, *в*, *г*).

TLA часто называют сдвоенным конвейером, поскольку он может быть разделен на два линейных конвейера типа BLA. Соответственно, TLA можно получить объединением двух BLA с одним общим потоком данных.

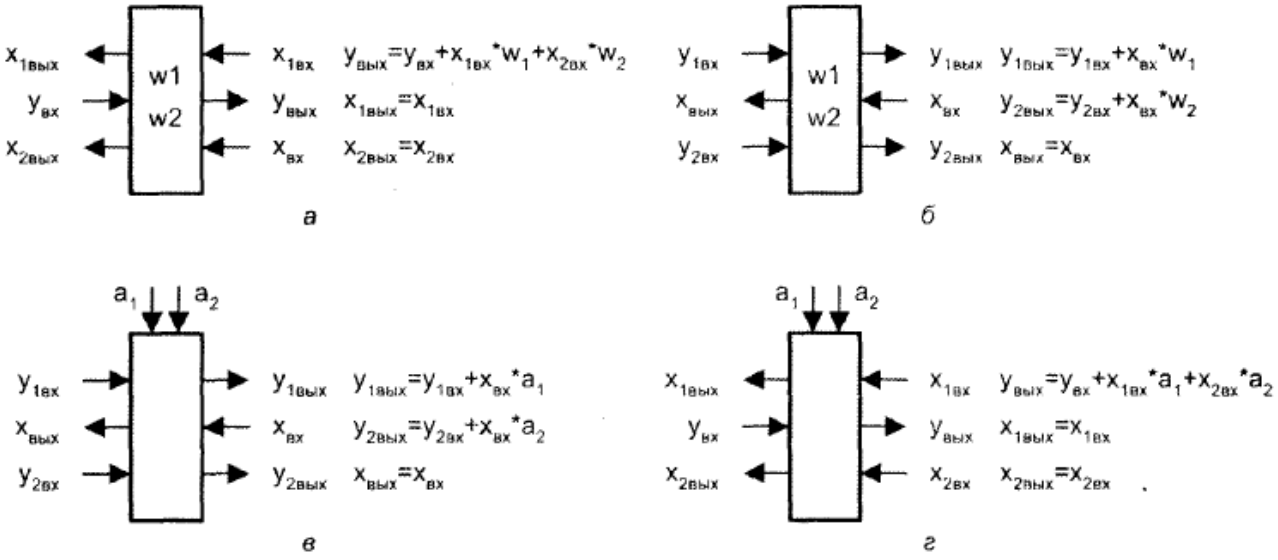


Рис. 1.31. Сдвоенные IPS-элементы: *а*, *б* – с хранимыми в памяти двумя значениями; *в*, *г* – с внешними данными

Представленные реализации алгоритма векторного произведения матриц выполняют эту операцию за одно и то же время, но в случае ULA в вычислениях участвуют вдвое меньше процессорных элементов. С другой стороны, ULA использует хранимые в памяти данные, на чтение и запись которых нужно какое-то время. В свою очередь, в схеме BLA требуется дополнительное время на операции ввода/вывода.

Структура процессорных элементов

Тип ПЭ выбирается в соответствии с назначением систолической матрицы и структурой пространственных связей. Наиболее распространены процессорные элементы, ориентированные на умножение с накоплением.

На рис. 1.32 показаны ПЭ для двух типов матриц: прямоугольной (см. рис, 1.32, *а*) и гексагональной (см. рис. 1.32, *б*).

В обоих случаях на вход ПЭ подаются два операнда – $A_{вх}$, $B_{вх}$, а выходят операнды $A_{вых}$, $B_{вых}$ и частичная сумма $C_{вых}$. На *n*-м шаге работы систолической системы ПЭ выполняет

$$C_{вых}^{(n)} = C_{вх}^{(n-1)} + A_{вх}^{(n-1)} B_{вх}^{(n-1)}$$

на основе операндов, полученных на $(n-1)$-м шаге, при этом операнды на входе и выходе ПЭ одинаковы:

$$A_{вых}^{(n)} = A_{вх}^{(n-1)},\ B_{вых}^{(n)} = B_{вх}^{(n-1)}.$$

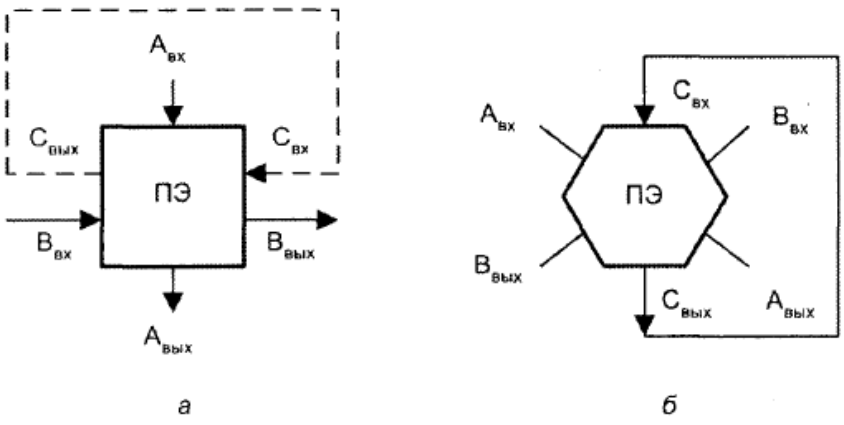


Рис. 1.32. Структура ПЭ: *а* – для прямоугольной систолической матрицы; *б* – для гексагональной систолической матрицы

Частичная сумма поступает на вход ПЭ либо с данного процессорного элемента (штриховая линия), либо с соседнего ПЭ матрицы.

1.7. Функционально распределенные и перестраиваемые вычислительные системы

Функционально распределенные вычислительные системы

На вычислительные системы общего назначения, используемые для научно-технических расчетов и моделирования, в системах автоматизированного проектирования и управления производством возлагается выполнение широкого спектра задач. При этом вычислительная система должна реализовать обширный набор функций над различными типами и структурами данных: обработку целочисленных значений, действительных чисел, графической информации и изображений, текстов, матричную обработку, трансляцию программ, доступ к данным, организованным в наборы или базы, и т. д. Кроме того, для управления вычислительными процессами и функционированием необходимо реализовать специфические функции управляющих программ операционной системы, управления виртуальной памятью, средствами ввода – вывода и передачи данных, а также контроль и диагностику системы и др.

В ЭВМ первого и второго поколений все эти функции реализовались одним процессором и интерпретировались им в виде арифметических и логических операций. Такое же положение сохранилось в основном и в ЭВМ третьего поколения, в которых лишь простейшие операции ввода – вывода снимались с центрального процессора и передавались специализированным устройствам – каналам или процессорам ввода – вывода.

Высокопроизводительные системы общего назначения создаются на основе многопроцессорных комплексов. Использование в таких системах однотипных процессоров, аналогичных процессорам ЭВМ общего назначения, оказывается неэкономичным, поскольку в каждом процессоре в каждый момент времени используется лишь часть ресурсов, обеспечивающих обработку данных одного типа. Наиболее экономичный способ построения многопроцессорной системы общего назначения – использование специализированных процессоров, ориентированных на реализацию определенных функций – обработку скалярных величин, текстов, матричную обработку, трансляцию программ, управление данными и др. При этом значительно сокращаются затраты на оборудование в процессоре и повышается его производительность. Кроме того, совокупность таких процессоров предоставляет необходимый для решения задач набор функций, который можно изменять, по-разному комплектуя систему и приспосабливая ее к рабочей нагрузке.

Многопроцессорные вычислительные системы, построенные на основе разнотипных процессоров, ориентированных на реализацию опреде-

ленных функций, называются функционально распределенными вычислительными системами (ФРВС). Это неоднородные системы, и строятся они как проблемно-ориентировочные – путем включения в их состав набора процессоров, соответствующего потребностям обрабатываемых задач.

Структура и функционирование

Принцип структурной организации ФРВС представлен па рис. 1.33. Система состоит из совокупности процессоров, имеющих индивидуальную память, и основной памяти. Ядро системы обеспечивает информационное сопряжение всех устройств. Ядро может быть реализовано в виде системной шины (магистрали), коммутационного поля или коммутатора основной памяти. В первых двух случаях каждый процессор может обмениваться данными с любыми другими процессорами и основной памятью. При использовании коммутатора основной памяти обмен данными производится только через память.

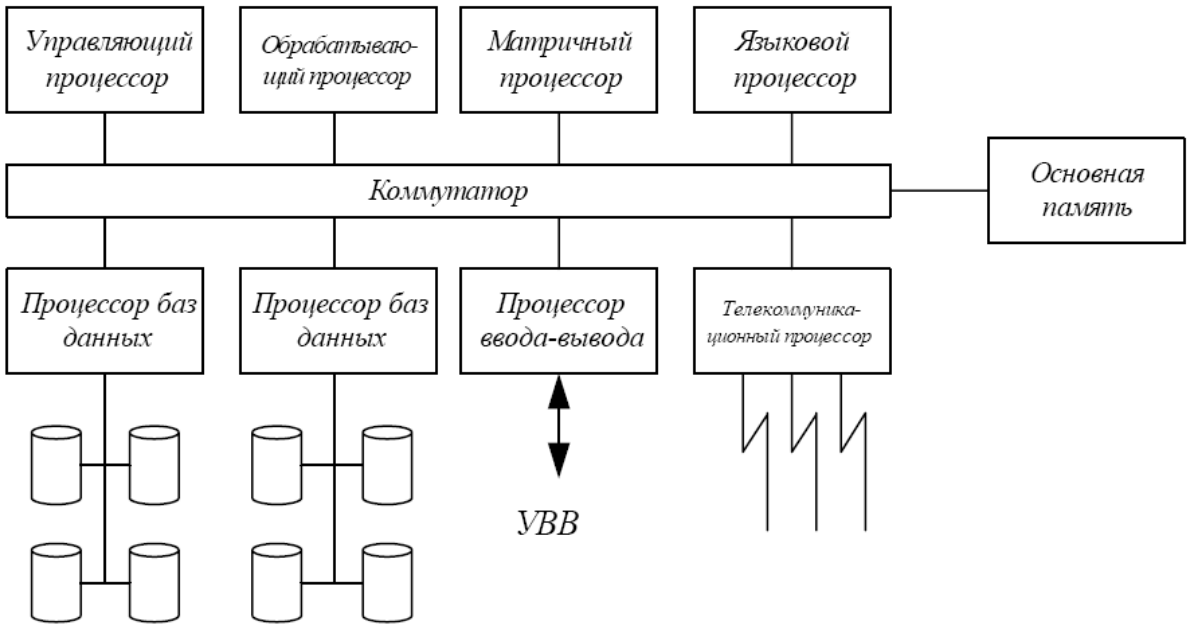


Рис. 1.33. Функционально распределенная вычислительная система

В представленной структуре управляющий процессор реализует супервизорные функции – управление ресурсами и задачами, обрабатывающий процессор – обработку числовых и символьных данных, матричный процессор – матричную и векторную обработку, языковой процессор – трансляцию программ, процессоры баз данных – доступ к наборам данных и управление базами данных, процессор ввода – вывода обслуживает уст-

ройства ввода – вывода и телекоммуникационный процессор обеспечивает передачу данных по каналам связи. Состав процессоров в конкретной системе зависит от класса решаемых задач. Так, в системе могут использоваться два обрабатывающих процессора или несколько телекоммуникационных.Обработка каждой задачи распределяется между процессорами. При этом функции управления данными реализуются процессорами, изображенными в нижней части рисунка. Разные шаги заданий, программы и ветви (блоки) программ выполняются обрабатывающим, матричным и языковым процессорами. Распределение ресурсов между задачами и управление задачами производится управляющим процессором, который реализует управляющие программы операционной системы. Загрузка оборудования увеличивается за счет мультипрограммирования и, возможно, параллельных вычислений на уровне подзадач.

Специализация процессоров обеспечивается на разных уровнях – на уровне структуры, микропрограммном и программном. Специализация на уровне структуры достигается за счет использования и операционной части процессора специальных регистровых структур и микроопераций, эффективно реализующих заданный набор операций. Такими являются матричные процессоры, содержащие совокупность арифметическо-логических устройств, с помощью которых параллельно обрабатываются векторы и матрицы. Специализация на микропрограммном уровне сводится к созданию с помощью микропрограмм специализированного набора операций, ориентированного на вычисление заданного набора функций. При использовании ОЗУ для хранения микропрограмм специализация процессора производится путем динамического микропрограммирования – загрузки в память соответствующего набора микропрограмм. В этом случае можно оперативно изменять конфигурацию системы, загружая в структурно одинаковые процессоры необходимые наборы микропрограмм. Функциональная специализация процессоров на программном уровне достигается за счет загрузки в процессор соответствующего набора программ.

В ФРВС используются, как правило, все три уровня специализации процессоров. Обрабатывающие и матричные процессоры имеют специализированную структуру: первые – для выполнения традиционных операций над логическими значениями, целыми и действительными числами и строками символов, последние – для производства векторных и матричных операций. Остальные процессоры функционально специализируются на уровне микропрограмм или программ.

Реализация

Принцип функционально распределенной организации применялся уже в ЭВМ третьего поколения, где для ввода – вывода использовались каналы и процессоры ввода – вывода. Примерами являются ЭВМ серий IBM 360, IBM 370 и ЕС ЭВМ.

Одна из первых систем, в полной мере реализующая принцип функционально распределенной организации, – система SYMBOL, созданная в 1970 г. Это неоднородная восьмипроцессорная система (рис. 1.34).

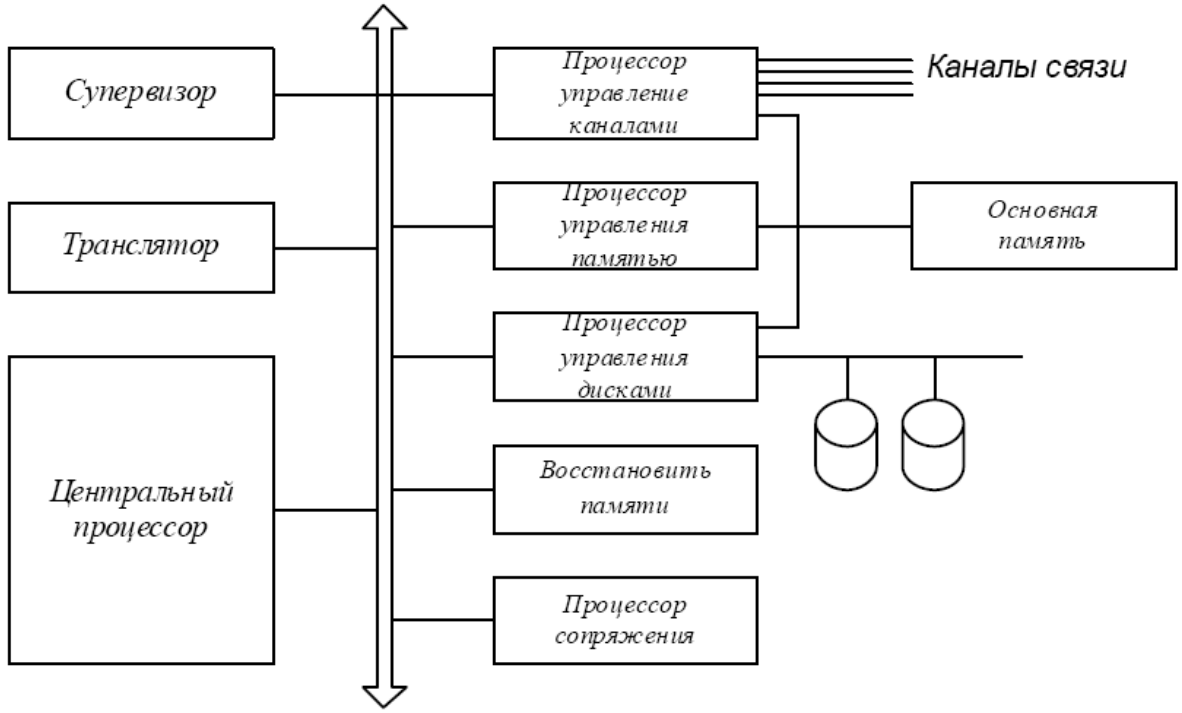


Рис. 1.34. Система SYMBOL

Процессор-супервизор управляет работой всей системы, координируя остальные процессоры, создавая очереди заявок к ним и распределяя процессоры между задачами. Процессор-транслятор обеспечивает перевод операторов с входного языка на внутренний язык системы (на машинный язык). Центральный процессор реализует обычные функции выборки команд и также арифметические и логические операции. В системе используется виртуальная память, работа которой обеспечивается процессором управления памятью. Этот процессор обрабатывает заявки от других процессоров на запись и чтение данных произвольной структуры. Виртуальная память состоит из 216 страниц, содержащих 256 64-битных слов, емкость основной памяти – 8 кслов (32 страницы). В дисковой памяти раз-

мещается примерно 50 тыс. страниц. Распределение емкости дисковой памяти, поиск и передача требуемой страницы данных реализуются процессором управления дисками. Управление вводом – выводом данных возложено на процессор управления каналами. К этому процессору через каналы передачи данных подключены внешние устройства. Редактирование и преобразование вводимых – выводимых данных обеспечивается процессором сопряжения, который работает в основном совместно с процессором управления каналами.

Процессоры сопрягаются посредством главной шины, состоящей из 111 линий, причем используются для передачи слова данных 64 линии, адреса слова – 24 линии, адреса абонента, которому направляются данные, – 5 линий. Остальные линии служат для передачи кода операции, приоритета сообщений и для синхронизации работы абонентов. При относительно низком быстродействии (длительность цикла процессоров – 320 нс и оперативной памяти в 2,5 мкс) система отличается высокой производительностью, составляющей 75 тыс. операторов входного языка в минуту, что примерно в 10 раз больше, чем у больших ЭВМ общего назначения.

На рис. 1.35 представлена упрощенная структурная схема вычислительной системы семейства System/80 фирмы IBM.

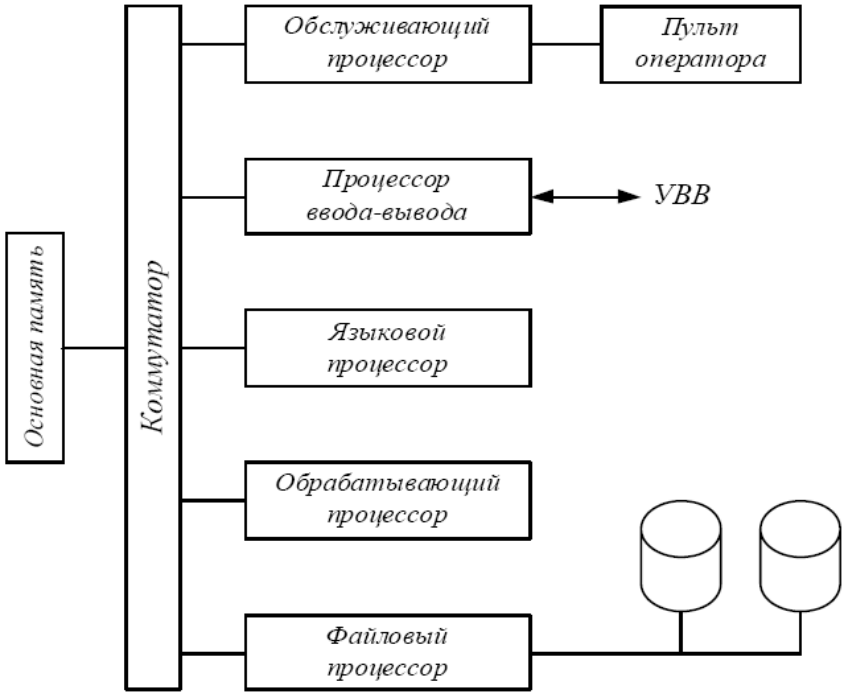


Рис. 1.35. Система System/80

В системе может использоваться до 8 – 16 процессоров, взаимодействующих через основную память и интерфейс прямого управления. Обслуживающий процессор обеспечивает работу пульта системы, планирование, контроль и диагностику. Процессор ввода – вывода обслуживает внешние устройства и выполняет первичную обработку (редактирование) вводимых – выводимых данных. Языковой процессор предназначен для трансляции программ с языка высокого уровня на машинный язык, т. е. в систему команд соответствующих процессоров. Обрабатывающий процессор выполняет обычные операции центрального процессора ЭВМ. Файловый процессор управляет данными, реализуя создание, открытие и закрытие наборов данных и доступ к данным, хранимым в наборах с различной организацией. Каждый процессор имеет собственную оперативную память. Функциональная ориентация процессоров обеспечивается на микропроцессорном уровне – путем загрузки в процессор соответствующих наборов микропрограмм. Возможности системы могут расширяться за счет подключения нескольких процессоров ввода – вывода, обрабатывающих процессоров, а также матричного процессора.

Системы с перестраиваемой структурой

Универсальный способ создания высокопроизводительных и высоконадежных вычислительных систем – объединение ЭВМ (процессоров) в многомашинные (многопроцессорные) комплексы, обеспечивающие:

1) параллелизм процессоров управления, доступа к данным и обработки;

2) распределенность процессоров управления, доступа к данным и обработки между модулями системы, т. е. децентрализированность управления работой системы и асинхронность взаимодействия процессоров и модулей;

3) перестраиваемость структуры с целью адаптации системы к потребностям задач в ресурсах и повышения устойчивости к отказам элементов;

4) открытость, т. е. возможность развития системы за счет подключения к ней дополнительных модулей без изменения принципов функционирования имеющихся модулей;

5) модульность технических и программных средств и регулярность (в пределе – однородность) структуры.

Параллелизм вычислительных процессов и процессов управления создает основу для повышения производительности системы. Распределенность процессов позволяет строить высокопроизводительные системы из достаточно простых модулей, например, из микро-ЭВМ с относительно небольшим быстродействием и ограниченной емкостью памяти. Перестраиваемость структуры обеспечивает, с одной стороны, высокую производительность системы за счет ее адаптации к вычислительным процессам и составу обрабатываемых задач и, с другой стороны, – живучесть системы при отказах элементов. Открытость системы позволяет в рамках фиксированной архитектуры создавать системы разной производительности за счет изменения числа модулей от единиц до десятков, сотен и, возможно, тысяч. Модульность технических и программных средств существенно упрощает разработку и производство элементов системы, за счет чего снижается ее стоимость, а также порождает регулярность структуры и, следовательно, упрощает управление системой (процессами и ресурсами) и ее эксплуатацию.

В последние десятилетия ведутся интенсивные исследования в области создания параллельных распределенных открытых многомодульных систем с перестраиваемой структурой, на основе которых разработано большое число экспериментальных и рабочих систем повышенной производительности и надежности. Архитектура систем с рассматриваемыми свойствами имеет особое значение для использования микро-ЭВМ в качестве элементной базы. Возможность неограниченного объединения микро-ЭВМ в системе, эффективно адаптирующихся к потребностям задач, позволила бы решить многие проблемы, в том числе – обеспечить пользователей высокопроизводительными средствами обработки числовых данных, графической информации и изображений. Однако для создания таких систем необходимо решить комплекс проблем системного управления, представляющих собой программно-аппаратурную надстройку над базовыми микро-ЭВМ. Среди этих проблем важнейшими являются:

1) структурная организация систем, обеспечивающая образование ансамблей процессоров, запоминающих устройств и каналов обмена данными, соответствующих потребностям вычислительного процесса, при умеренных затратах ресурсов на их организацию и координацию;

2) организация вычислительных процессов, обеспечивающая их параллелизм, распределение по системе модулей и координацию асинхронно выполняемых подпроцессов при умеренных издержках;

3) создание языков высокого уровня, описывающих алгоритмы в системно-независимой форме и сохраняющих представление о параллелизме вычислительного процесса.

К настоящему времени эти проблемы не решены окончательно и известные разработки вычислительных систем с перестраиваемой структурой лишь отчасти обладают требуемыми свойствами.

Структурная организация

Вычислительные системы с перестраиваемой структурой строятся на основе микропроцессорных модулей. Модуль должен реализовать следующие функции: 1) обработку данных, сводящуюся к обработке логических значений, числовых значений, представленных в виде целых и действительных чисел, и строк символов; 2) управление вычислительным процессом, обеспечивающее взаимодействие модуля с ансамблем модулей, реализующих процесс, и с системой в целом; 3) установление соединений с другими модулями и передачу данных между ними для обеспечения вычислительных процессов.

С учетом указанных функций модуль вычислительной системы рассматривается как совокупность трех процессоров (рис. 1.36): обрабатывающего (ОП), управляющего (УП) и коммутационного (коммуникационного) (КМ). Коммутационный процессор обеспечивает обслуживание нескольких (обычно двух – шести) каналов передачи данных. Физически модуль может реализоваться на основе одной микро-ЭВМ, выполняющей в мультипрограммном режиме функции обработки, управления процессами и передачи данных, или па основе нескольких микропроцессоров, между которыми разделяются вышеперечисленные функции.

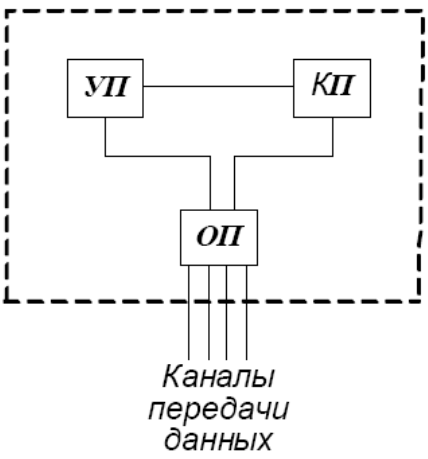


Рис. 1.36. Состав модуля системы с перестраиваемой структурой

В вычислительных системах с перестраиваемой структурой модули объединяются в простейшие структуры, позволяющие достаточно легко определять пути соединений между взаимодействующими модулями. Наиболее подходящими для построения рассматриваемых систем являются матричные, пирамидальные и кубические структуры. Фрагмент матричной структуры изображен на рис. 1.37. Модули, в которых выделен коммутационный процессор, соединяются посредством последнего в матрицу. Коммутационные процессоры и каналы связи образуют в совокупности коммутационное поле, обеспечивающее соединение взаимодействующих модулей и передачу данных между ними. Часть модулей системы специализируется на обслуживании периферийных устройств – накопителей на магнитных дисках и лентах и устройств ввода – вывода. Управляющие и коммутационные процессоры модулей ввода – вывода обеспечивают взаимодействие внешних устройств с процессами, реализуемыми в любом из модулей системы.

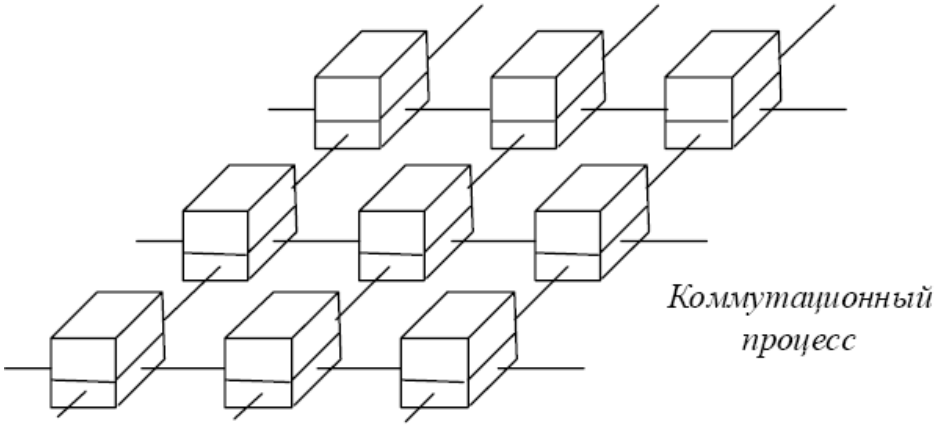


Рис. 1.37. Фрагмент матричной структуры

Рассмотрим другой способ структурной организации системы с перестраиваемой структурой на основе вычислительных комплексов, имеющих коммутационное поле, которое образовано совокупностью коммутаторов с децентрализованным управлением. Наиболее экономичными являются поля с многоуровневой организацией, пример которых приведен на рис. 1.38.

Здесь a_1,..., a_{16} – входы коммутатора, к которым подсоединяются микропроцессорные модули. Штриховыми линиями показаны примеры соединения модулей через коммутатор. В таких полях нижний уровень коммутации обеспечивает соединения между соседними модулями, образующими отдельные группы, следующий уровень коммутации – соединения между соседними группами и т. д.

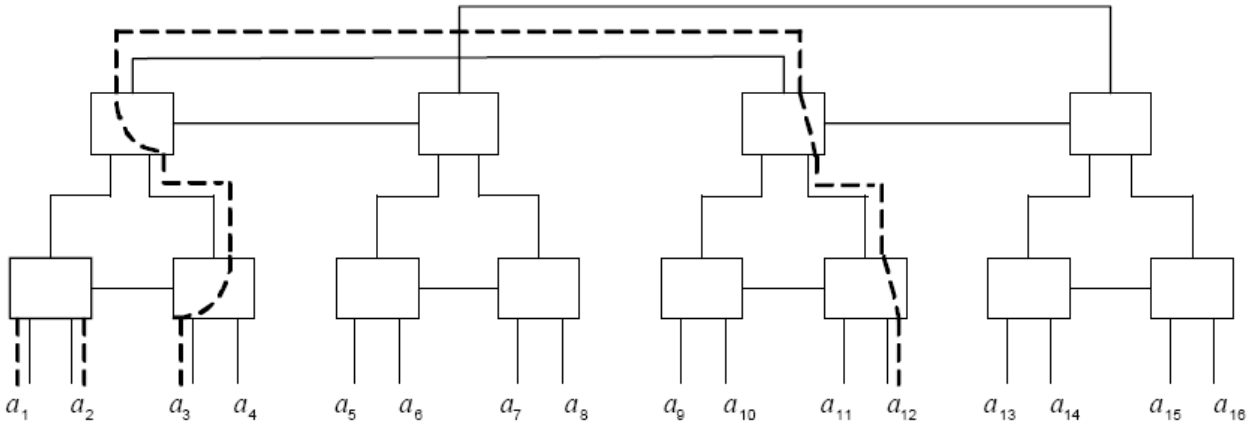


Рис. 1.38. Многоуровневое коммутационное поле

Многоуровневые коммутационные поля позволяют создавать соединения между любыми парами процессорных модулей с помощью умеренного числа коммутационных модулей в системах с матричной, кубической и пирамидальной структурами. Однако среднее число одновременно устанавливаемых соединений в многоуровневых коммутационных полях меньше среднего числа соединений, обеспечиваемых матричными и аналогичными структурами, что приводит к снижению степени параллелизма вычислительных процессов и, следовательно, к уменьшению производительности системы.

Организация вычислительных процессов

Основные проблемы организации вычислений в системах с перестраиваемой структурой связаны с обеспечением параллелизма вычислений и распределенного децентрализованного управлении процессами и ресурсами. Эти проблемы разработаны только в первом приближении, и известные способы организации параллельных вычислений в распределенных системах с децентрализованным управлением еще не достигли необходимого уровня универсальности и формализации.

Параллельная обработка задач, т. е. мультипрограммный режим функционирования системы, обеспечивается достаточно простыми средствами. После ввода задания в систему модуль, принявший задание, посылает через коммутационное поле запрос на поиск свободного обрабатывающего модуля. Когда свободный модуль найден, ему посылается задание, определяющее имена наборов данных, в которых размещается программа, исходные данные и в которые должны быть помещены результаты вычислений. Из задания и программы модуль получает сведения о ресурсах, необходимых для выполнения задания: емкости операционной памяти, числе

процессоров и неразделяемых наборах данных. Модуль закрепляет за собой необходимые ресурсы, и после обеспечения задания требуемыми ресурсами инициируется процесс выполнения задачи. По завершении обработки ресурсы освобождаются и в дальнейшем предоставляются очередным заданиям. Число процессов, реализуемых параллельно, определяется числом модулей, входящих в состав системы, и при наличии очереди заданий производительность системы пропорциональна числу модулей.

Параллельные программы строятся традиционными способами: выделением подзадач и ветвей программы, операций над векторами и матрицами и организацией конвейерной обработки данных. Наиболее просто реализуются вычисления с выделением подзадач и параллельных ветвей. При возникновении ветви в ведущей программе модуль посылает запрос на поиск свободного модуля, в который загружается программа и данные ветви, и ветвь выполняется как самостоятельная задача, по завершении которой в ведущий модуль отсылаются результаты обработки. Параллельные вычисления по конвейерной и матричной схемам организуются за счет создания соответствующих конфигураций связей между модулями – линейных (кольцевых) и матричных структур. Построение таких структур в многомодульных системах, в которых часть модулей занята выполнением ранее созданных задач, является пока нерешенной проблемой. Обычно для матричных вычислений в систему встраивается в качестве специального модуля матричный процессор, обеспечивающий высокопроизводительную обработку блоков данных.

В вычислительной системе с перестраиваемой структурой должно быть реализовано распределенное (децентрализованное) управление ресурсами. Это означает, что в системе не должно быть выделенного модуля (даже многократно зарезервированного), на который возложена задача централизованного управления функционированием системы. Распределенное управление основано на согласованной работе всех модулей системы, каждый из которых реализует одинаковый набор правил управления, обеспечивающий эффективное использование всех ресурсов системы. Распределенное управление повышает надежность системы, поскольку каждый модуль способен реализовать управление ресурсами и процессами, и одновременно повышает производительность системы, так как управляющие решения формируются без затрат времени на сбор информации о состоянии всех элементов системы (а за это время ситуация в системе может существенно измениться).

Как в любой многопроцессорной системе, механизм управления должен исключать взаимную блокировку процессов при запросах ресур-

сов. Такая ситуация возникает, если процесс A располагает ресурсом a и требует для своего исполнения ресурс b, а процесс B располагает ресурсом b и дополнительно нуждается в ресурсе a. Например, процесс A располагает модулем a и требует соединения b с ним, а процесс B располагает соединением b и требует модуль a. Для предотвращения блокировок используются различные механизмы управления ресурсами, одновременное формирование запросов на все необходимые ресурсы и освобождение всех ресурсов, если не выделен хотя бы одни из них; разделение ресурсов по типам и иерархический порядок выделения ресурсов и т. д. Кроме того, механизм управления должен принимать решения о передислокации программ и наборов данных между модулями, о необходимости подключения к процессу дополнительных модулей или последовательной реализации алгоритмически параллельных процессов на одном модуле и, наконец, о распределении задач между модулями, обеспечивающем необходимое время решения и высокую производительность системы. Для управления процессами обычно предлагаются эвристические процедуры, не требующие большой емкости памяти и трудоемких вычислений. Однако эффективность большинства предлагаемых процедур к настоящему времени не оценена в достаточной степени.

Вопросы и задания для самопроверки

1. Дайте определение понятию «параллелизм».
2. Какие формы параллелизма вы знаете?
3. Дайте определение многомашинного вычислительного комплекса.
4. На какие типы по характеру связей можно разделить многомашинные вычислительные комплексы?
5. Дайте определение многопроцессорного вычислительного комплекса.
6. Какие типы структурной организации многопроцессорных вычислительных комплексов вы знаете?
7. В чем заключается принцип конвейеризации ЭВМ?
8. Какова основная область применения матричных процессоров?
9. Опишите основные типы архитектурной организации массива процессорных элементов в матричной ВС.
10. Опишите принцип работы ассоциативного запоминающего устройства.
11. Каковы особенности работы вычислительной системы с систолической структурой?
12. В чем заключаются особенности работы функционально распределенных вычислительных систем?

МОДУЛЬ 2. КОМПЬЮТЕРНЫЕ СЕТИ

Цель модуля – приобретение теоретических и практических основ организации и функционирования компьютерных сетей, знакомство с современными компьютерными сетевыми технологиями.

В результате изучения модуля студенты **должны:**

- знать принципы построения и функционирования вычислительных сетей, их характеристики и параметры;
- знать классификацию вычислительных систем и сетей, особенности современных технологий и архитектур вычислительных комплексов и сетей, аппаратное и программное обеспечение компьютерных сетей;
- иметь представление о методах проектирования локальных сетей для решения конкретных практических задач, перспективах и тенденциях развития современных сетевых технологий;
- ознакомиться со структурообразующим оборудованием физического, канального и сетевого уровней вычислительных сетей;
- уметь осуществлять конфигурирование сетевых аппаратных средств в современных операционных системах;
- иметь представление о методах управления доступом в вычислительных сетях.

Содержание модуля:

2.1. Эволюция вычислительных систем. Основные программные и аппаратные компоненты сети.

2.2. Понятие «открытая система». Модель OSI.

2.3. Стандартные стеки коммуникационных протоколов.

2.4. Линии связи. Типы, аппаратура, характеристики линий связи.

2.5. Стандарты кабелей.

2.6. Методы передачи дискретных данных на физическом уровне.

2.7. Методы передачи данных канального уровня.

2.8. Методы коммутации: коммутация каналов и коммутация пакетов.

2.9. Протоколы и стандарты локальных сетей. Структура стандартов IEEE 802.X.

2.10. Протокол LLC уровня управления логическим каналом.

2.11. Технология Ethernet (802.3). Метод доступа CSMA/CD.

2.12. Спецификации физической среды Ethernet. Стандарт 10Base-5.

2.13. Технология Ethernet. Стандарты 10Base-2, 10Base-T, 10Base-F. Понятие домена коллизий.

2.14. Технология Token Ring (802.5).

2.15. Технология FDDI.

2.16. Технология Fast Ethernet.

2.17. Высокоскоростная технология Gigabit Ethernet.

2.18. Структурированная кабельная система.

2.19. Сетевые адаптеры и концентраторы.

2.20. Логическая структуризация сети с помощью мостов и коммутаторов.

2.21. Принципы работы мостов.

2.22. Коммутаторы локальных сетей.

2.23. Техническая реализация коммутаторов.

2.24. Характеристики, влияющие на производительность коммутаторов, дополнительные функции.

2.25. Виртуальные локальные сети. Типовые схемы применения коммутаторов в локальных сетях.

2.26. Принципы объединения сетей на основе протоколов сетевого уровня.

2.27. Адресация в IP-сетях.

2.28. Протокол IP.

2.29. Протоколы маршрутизации в IP-сетях.

2.30. Основные характеристики маршрутизаторов. Стирание граней между марщрутизаторами и коммутаторами.

Вопросы и задания для самопроверки

2.1. Эволюция вычислительных систем. Основные программные и аппаратные компоненты сети

Концепция вычислительных сетей является логическим результатом эволюции компьютерной технологии. Первые компьютеры 50-х годов – большие, громоздкие и дорогие – предназначались для очень небольшого числа избранных пользователей. Такие компьютеры не были предназначены для интерактивной работы пользователя, а использовались в режиме пакетной обработки.

Системы пакетной обработки

Системы пакетной обработки, как правило, строились на базе мэйнфрейма – мощного и надежного компьютера универсального назначения. Пользователи подготавливали перфокарты, содержащие данные и команды

программ, и передавали их в вычислительный центр. Операторы вводили эти карты в компьютер, а распечатанные результаты пользователи получали обычно только на следующий день (рис. 2.1). Таким образом, одна неверно набитая карта означала как минимум суточную задержку.

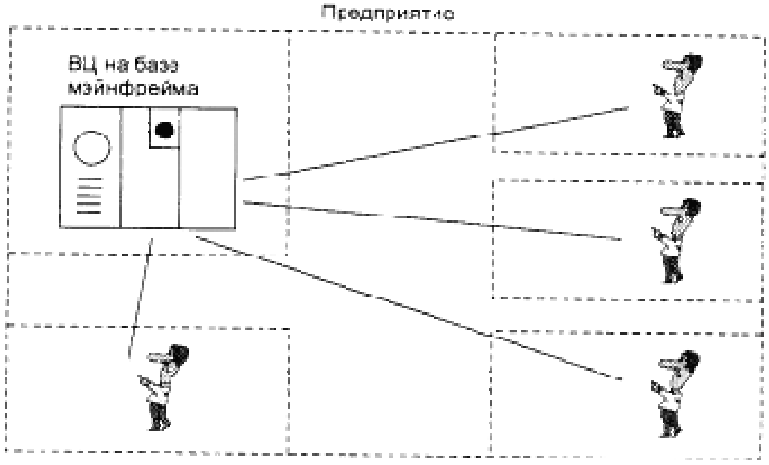


Рис. 2.1. Централизованная система на базе мэйнфрейма

Конечно, для пользователей интерактивный режим работы, при котором можно с терминала оперативно руководить процессом обработки своих данных, был бы гораздо удобней. Но интересами пользователей на первых этапах развития вычислительных систем в значительной степени пренебрегали, поскольку пакетный режим – это самый эффективный режим использования вычислительной мощности, так как он позволяет выполнить в единицу времени больше пользовательских задач, чем любые другие режимы. Во главу угла ставилась эффективность работы самого дорогого устройства вычислительной машины – процессора, в ущерб эффективности работы использующих его специалистов.

Многотерминальные системы – прообраз сети

По мере удешевления процессоров в начале 60-х годов появились новые способы организации вычислительного процесса, которые позволили учесть интересы пользователей. Начали развиваться интерактивные многотерминальные системы разделения времени (рис. 2.2). В таких системах компьютер отдавался в распоряжение сразу нескольким пользователям. Каждый пользователь получал в свое распоряжение терминал, с помощью которого он мог вести диалог с компьютером. Причем время реак-

ции вычислительной системы было достаточно мало для того, чтобы пользователю была не слишком заметна параллельная работа с компьютером и других пользователей. Разделяя таким образом компьютер, пользователи получили возможность за сравнительно небольшую плату пользоваться преимуществами компьютеризации.

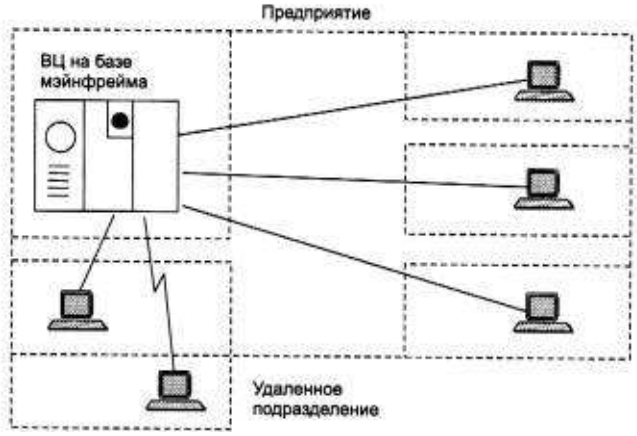


Рис. 2.2. Многотерминальная система - прообраз вычислительной сети

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

Таким образом, многотерминальные системы, работающие в режиме разделения времени, стали первым шагом на пути создания локальных вычислительных сетей. Но до появления локальных сетей нужно было пройти еще большой путь, так как многотерминальные системы хотя и имели внешние черты распределенных систем, все еще сохраняли централизованный характер обработки данных.

Появление глобальных сетей

К определенному времени назрела потребность в соединении компьютеров, находящихся на большом расстоянии друг от друга. Началось все с решения более простой задачи – доступа к компьютеру с терминалов, удаленных от него на многие сотни, а то и тысячи километров. Терминалы соединялись с компьютерами через телефонные сети с помощью модемов. Такие сети позволяли многочисленным пользователям получать удаленный доступ к разделяемым ресурсам нескольких мощных компьютеров класса суперЭВМ. Затем появились системы, в которых наряду с удаленными соединениями типа терминал – компьютер были реализованы и удаленные связи типа компьютер – компьютер. Компьютеры получили возможность обмениваться данными в автоматическом режиме, что, собственно, и является базовым механизмом любой вычислительной сети. С использованием этого механизма в первых сетях были реализованы службы обмена файлами, синхронизации баз данных, электронной почты и другие, ставшие теперь традиционными, сетевые службы.

Таким образом, хронологически первыми появились глобальные вычислительные сети. Именно при построении глобальных сетей были впервые предложены и отработаны многие основные идеи и концепции современных вычислительных сетей. Такие, например, как многоуровневое построение коммуникационных протоколов, технология коммутации пакетов, маршрутизация пакетов в составных сетях.

Первые локальные сети

В начале 70-х годов произошел технологический прорыв в области производства компьютерных компонентов – появились большие интегральные схемы. Их сравнительно невысокая стоимость и высокие функциональные возможности привели к созданию мини-компьютеров, которые стали реальными конкурентами мэйнфреймов.

Даже небольшие подразделения предприятий получили возможность покупать для себя компьютеры. Мини-компьютеры выполняли задачи управления технологическим оборудованием, складом и другие задачи уровня подразделения предприятия. Таким образом, появилась концепция распределения компьютерных ресурсов по всему предприятию. Однако при этом все компьютеры одной организации по-прежнему продолжали работать автономно (рис. 2.3).

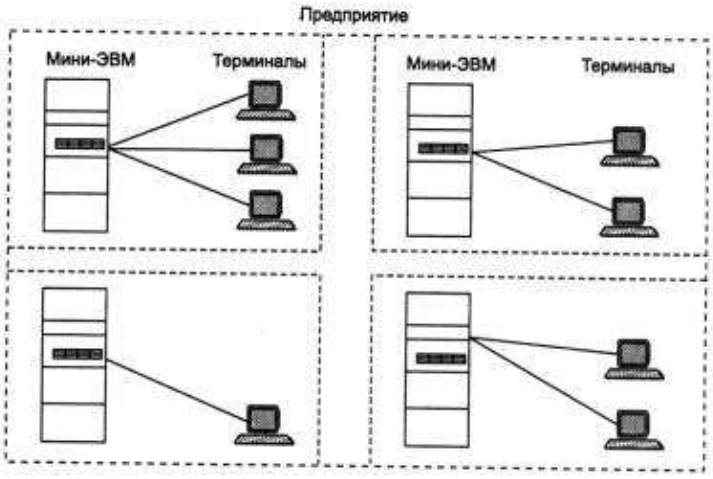


Рис. 2.3. Автономное использование нескольких мини-компьютеров на одном предприятии

Шло время, потребности пользователей вычислительной техники росли, им стало недостаточно собственных компьютеров, им уже хотелось получить возможность обмена данными с другими близко расположенными компьютерами. В ответ на эту потребность предприятия и организации стали соединять свои мини-компьютеры вместе и разрабатывать программное обеспечение, необходимое для их взаимодействия. В результате появились первые локальные вычислительные сети (рис. 2.4). Они еще во многом отличались от современных локальных сетей, в первую очередь – своими устройствами сопряжения.

На первых порах для соединения компьютеров друг с другом использовались самые разнообразные нестандартные устройства со своим способом представления данных на линиях связи, своими типами кабелей и т. п.

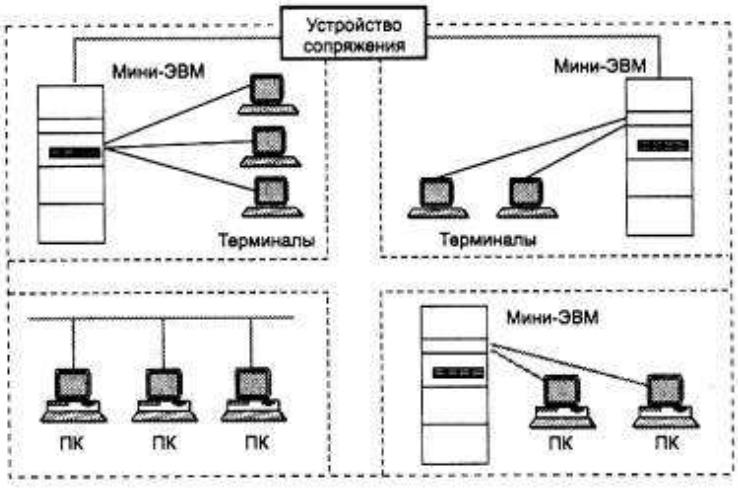


Рис. 2.4. Различные типы связей в первых локальных сетях

Создание стандартных технологий локальных сетей

В середине 80-х годов положение дел в локальных сетях стало кардинально меняться. Утвердились стандартные технологии объединения компьютеров в сеть – Ethernet, Arcnet, Token Ring. Мощным стимулом для их развития послужили персональные компьютеры.

Поэтому персональные компьютеры стали преобладать в локальных сетях, причем не только в качестве клиентских компьютеров, но и в качестве центров хранения и обработки данных, т. е. сетевых серверов, потеснив с этих привычных ролей мини-компьютеры и мэйнфреймы.

Для создания сети достаточно было приобрести сетевые адаптеры соответствующего стандарта, например Ethernet, стандартный кабель, присоединить адаптеры к кабелю стандартными разъемами и установить на компьютер одну из популярных сетевых операционных систем, например, NetWare. После этого сеть начинала работать, и присоединение каждого нового компьютера не вызывало никаких проблем, естественно, если на нем был установлен сетевой адаптер той же технологии.

Локальные сети в сравнении с глобальными сетями внесли много нового в способы организации работы пользователей.

Экономное расходование пропускной способности каналов связи часто являлось основным критерием эффективности методов передачи данных в глобальных сетях. В этих условиях различные процедуры прозрачного доступа к удаленным ресурсам, стандартные для локальных сетей, для глобальных сетей долго оставались непозволительной роскошью.

Современные тенденции

Сегодня вычислительные сети продолжают развиваться, причем достаточно быстро. Разрыв между локальными и глобальными сетями постоянно сокращается, во многом из-за появления высокоскоростных территориальных каналов связи, не уступающих по качеству кабельным системам локальных сетей. В глобальных сетях появляются службы доступа к ресурсам, такие же удобные и прозрачные, как и службы локальных сетей.

Изменяются и локальные сети. Вместо соединяющего компьютеры пассивного кабеля в них в большом количестве появилось разнообразное коммуникационное оборудование – коммутаторы, маршрутизаторы, шлюзы. Благодаря такому оборудованию появилась возможность построения больших корпоративных сетей, насчитывающих тысячи компьютеров и имеющих сложную структуру.

Основные программные и аппаратные компоненты сети

Даже в результате достаточно поверхностного рассмотрения работы в сети становится ясно, что вычислительная сеть – это сложный комплекс взаимосвязанных и согласованно функционирующих программных и аппаратных компонентов. Изучение сети в целом предполагает знание принципов работы ее отдельных элементов:

- компьютеров;
- коммуникационного оборудования;
- операционных систем;
- сетевых приложений.

Весь комплекс программно-аппаратных средств сети может быть описан многослойной моделью. В основе любой сети лежит аппаратный слой стандартизованных компьютерных платформ. В настоящее время в сетях широко и успешно применяются компьютеры различных классов – от персональных компьютеров до мэйнфреймов и суперЭВМ. Набор компьютеров в сети должен соответствовать набору разнообразных задач, решаемых сетью.

Второй слой – это коммуникационное оборудование. Хотя компьютеры и являются центральными элементами обработки данных в сетях, в последнее время не менее важную роль стали играть коммуникационные устройства. Кабельные системы, повторители, мосты, коммутаторы, маршрутизаторы и модульные концентраторы из вспомогательных компонентов сети превратились в основные наряду с компьютерами и системным программным обеспечением как по влиянию на характеристики сети, так и по стоимости. Сегодня коммуникационное устройство может представлять собой сложный специализированный мультипроцессор, который нужно конфигурировать, оптимизировать и администрировать. Изучение принципов работы коммуникационного оборудования требует знакомства с большим количеством протоколов, используемых как в локальных, так и глобальных сетях.

Третьим слоем, образующим программную платформу сети, являются операционные системы (ОС). От того, какие концепции управления локальными и распределенными ресурсами положены в основу сетевой ОС, зависит эффективность работы всей сети. При проектировании сети важно учитывать, насколько просто данная операционная система может взаимодействовать с другими ОС сети, насколько она обеспечивает безопасность и защищенность данных, до какой степени она позволяет наращивать число пользователей, можно ли перенести ее на компьютер другого типа и многие другие соображения.

Самым верхним слоем сетевых средств являются различные сетевые приложения, такие как сетевые базы данных, почтовые системы, средства архивирования данных, системы автоматизации коллективной работы и др. Очень важно представлять диапазон возможностей, предоставляемых приложениями для различных областей применения, а также знать, насколько они совместимы с другими сетевыми приложениями и операционными системами.

2.2. Понятие «открытая система». Модель OSI

В широком смысле открытой системой может быть названа любая система (компьютер, вычислительная сеть, ОС, программный пакет, другие аппаратные и программные продукты), которая построена в соответствии с открытыми спецификациями. Под термином «спецификация» (в вычислительной технике) понимают формализованное описание аппаратных или программных компонентов, способов их функционирования, взаимодействия с другими компонентами, условий эксплуатации, ограничений и особых характеристик. Понятно, что не всякая спецификация является стандартом. В свою очередь, под открытыми спецификациями понимаются опубликованные, общедоступные спецификации, соответствующие стандартам и принятые в результате достижения согласия после всестороннего обсуждения всеми заинтересованными сторонами.

В начале 80-х годов ряд международных организаций по стандартизации – ISO, ITU-T и некоторые другие – разработали модель, которая сыграла значительную роль в развитии сетей. Эта модель называется моделью взаимодействия открытых систем (*Open System Interconnection, OSI*) или моделью OSI. Модель OSI определяет различные уровни взаимодействия систем, дает им стандартные имена и указывает, какие функции должен выполнять каждый уровень. Модель OSI была разработана на основании большого опыта, полученного при создании компьютерных сетей, в основном глобальных, в 70-е годы. Полное описание этой модели занимает более 1000 страниц текста.

В модели OSI (рис. 2.5) 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.

Модель OSI описывает только системные средства взаимодействия, реализуемые операционной системой, системными утилитами, системными аппаратными средствами. Модель не включает средства взаимодействия приложений конечных пользователей. Свои собственные протоколы взаимодействия приложения реализуют, обращаясь к системным средст-

вам. Поэтому необходимо различать уровень взаимодействия приложений и прикладной уровень.

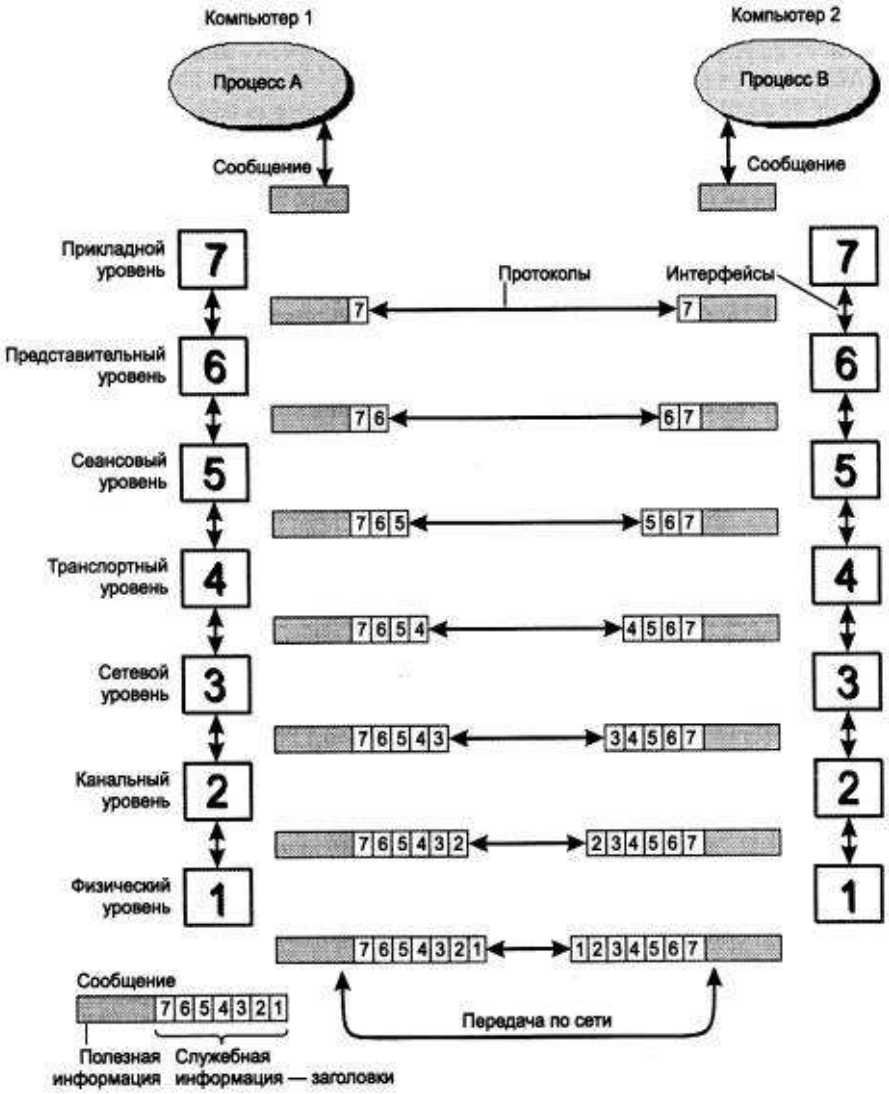


Рис. 2.5. Модель взаимодействия открытых систем ISO/OSI

Следует 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 OSI.

Итак, пусть приложение обращается с запросом к прикладному уровню, например, к файловой службе. На основании этого запроса про-

граммное обеспечение 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 случае 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. Но для того, чтобы доставить эту информацию по назначению, предстоит решить еще много задач, ответственность за которые несут нижележащие уровни.

После формирования сообщения прикладной уровень направляет его вниз по стеку 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 достигает нижнего, физического уровня, который, собственно, и передает его по линиям связи машине-адресату. К этому моменту сообщение «обрастает» заголовками всех уровней (рис. 2.6).

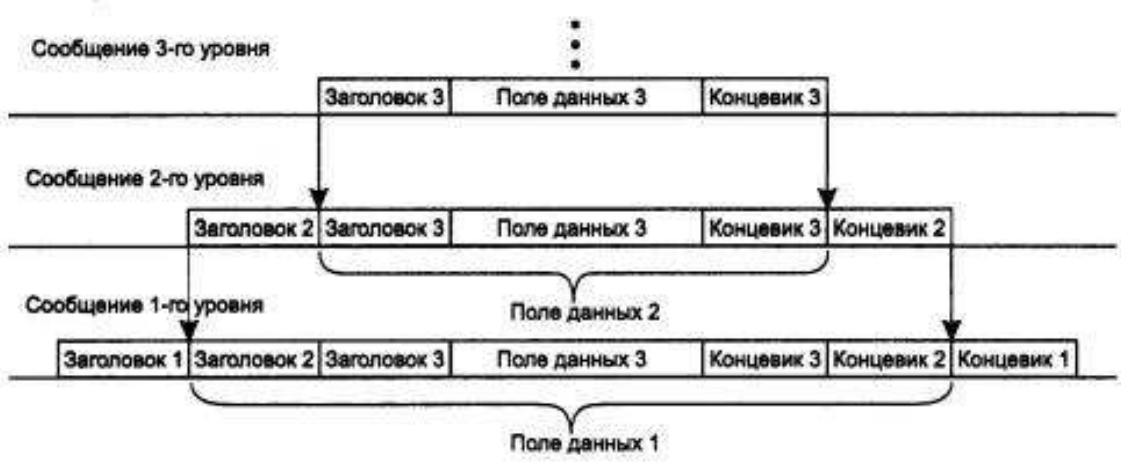


Рис. 2.6. Вложенность сообщений различных уровней

Когда сообщение по сети поступает на машину-адресат, оно принимается ее физическим уровнем и 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 передает сообщение вышележащему уровню.

Наряду с термином *сообщение (message)* существуют и другие термины, применяемые сетевыми специалистами для обозначения единиц данных в процедурах обмена. В стандартах ISO для обозначения единиц данных, с которыми имеют дело протоколы разных уровней, используется общее название *протокольный блок данных (Protocol Data Unit, PDU)*. Для обозначения блоков данных определенных уровней часто используются специальные названия – *кадр (frame), пакет (packet), дейтаграмма (datagram), сегмент (segment).*

В модели OSI различаются два основных типа протоколов. В протоколах *с установлением соединения (connection-oriented)* 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. Телефон – это пример взаимодействия, основанного на установлении соединения.

Вторая группа протоколов – протоколы *без предварительного установления соединения (connectionless)*. Такие протоколы называются также *дейтаграммными* протоколами. Отправитель просто передает сообщение, когда оно готово. Опускание письма в почтовый ящик – это пример связи без предварительного установления соединения. При взаимодействии компьютеров используются 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 цифровой территориальный канал. К этому уровню имеют отношение характеристики физических сред передачи данных, такие как полоса пропускания, помехозащищенность, волновое сопротивление и другие. На этом же уровне определяются характеристики электрических сигналов, передающих дискретную информацию, например, крутизна фронтов импульсов, уровни напряжения или тока передаваемого сигнала, тип кодирования, скорость передачи сигналов. Кроме этого, здесь стандартизуются типы разъемов и назначение каждого контакта.

Функции физического уровня реализуются во всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом.

Примером протокола физического уровня может служить спецификация l0-Base-T технологии Ethernet, которая определяет в качестве используемого кабеля неэкранированную витую пару категории 3 с волновым сопротивлением 100 Ом, разъем RJ-45, максимальную длину физического сегмента 100 метров, манчестерский код для представления данных в кабеле, а также некоторые другие характеристики среды и электрических сигналов.

Канальный уровень

На физическом уровне просто пересылаются биты. При этом не учитывается, что в некоторых сетях, в которых линии связи используются (разделяются) попеременно несколькими парами взаимодействующих компьютеров, физическая среда передачи может быть занята. Поэтому одной из задач канального уровня (*Data Link layer*) является проверка доступности среды передачи. Другой задачей канального уровня является реализация механизмов обнаружения и коррекции ошибок. Для этого на канальном уровне биты группируются в наборы, называемые кадрами (*frames*). Канальный уровень обеспечивает корректность передачи каждого кадра, помещая специальную последовательность бит в начало и конец каждого кадра, для его выделения, а также вычисляет контрольную сумму, обрабатывая все байты кадра определенным способом и добавляя контрольную сумму к кадру. Когда кадр приходит по сети, получатель снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой из кадра. Если они совпадают, кадр считается правильным и принимается. Если же контрольные суммы 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 протоколах канального уровня, используемых в локальных сетях, заложена определенная структура связей между компьютерами и способы их адресации. Хотя канальный уровень и обеспечивает доставку кадра между любыми двумя узлами локальной сети, он это делает только в сети с совершенно определенной топологией связей, именно той топологией, 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 из них с помощью мостов и коммутаторов. Примерами протоколов канального уровня являются протоколы Ethernet, Token Ring, FDDI, l00VG-AnyLAN.

В локальных сетях протоколы канального уровня используются компьютерами, мостами, коммутаторами и маршрутизаторами. В компьютерах функции канального уровня реализуются совместными усилиями сетевых адаптеров и их драйверов.

Тем не менее, для обеспечения качественной транспортировки сообщений в сетях любых топологий и технологий функций канального уровня оказывается недостаточно, поэтому в модели OSI решение этой задачи возлагается на два следующих уровня – сетевой и транспортный.

Сетевой уровень

Сетевой уровень (*Network layer*) служит для образования единой транспортной системы, объединяющей несколько сетей, причем эти сети могут использовать совершенно различные принципы передачи сообщений между конечными узлами и обладать произвольной структурой связей. Функции сетевого уровня достаточно разнообразны. Начнем их рассмотрение на примере объединения локальных сетей.

Протоколы канального уровня локальных сетей обеспечивают доставку данных между любыми узлами только в сети с соответствующей типовой топологией, например, топологией иерархической звезды. Это очень жесткое ограничение, которое не позволяет строить сети с развитой структурой, например, сети, объединяющие несколько сетей предприятия в единую сеть, или высоконадежные сети, в которых существуют избыточные связи между узлами. Можно было бы усложнять протоколы канального уровня для поддержания петлевидных избыточных связей, но принцип разделения обязанностей между уровнями приводит к другому решению. Чтобы, с одной стороны, сохранить простоту процедур передачи данных для типовых топологий, а, с другой, допустить использование произвольных топологий, вводится дополнительный сетевой уровень.

На сетевом уровне сам термин «сеть» наделяют специфическим значением. В данном случае под сетью понимается 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 соединяются между собой специальными устройствами, называемыми маршрутизаторами. *Маршрутизатор* –

это устройство, которое собирает информацию о топологии межсетевых соединений и на ее основании пересылает пакеты сетевого уровня в сеть назначения. Чтобы передать сообщение от отправителя, находящегося в одной сети, получателю, находящемуся в другой сети, нужно совершить некоторое количество транзитных передач между сетями, или хопов (от *hop* – прыжок), каждый раз выбирая подходящий маршрут. Таким образом, маршрут представляет собой последовательность маршрутизаторов, через которые проходит пакет.

На рис. 2.7 показаны четыре сети, связанные тремя маршрутизаторами. Между узлами A и B данной сети пролегают два маршрута: первый – через маршрутизаторы 1 и 3, а второй – через маршрутизаторы 1, 2 и 3.

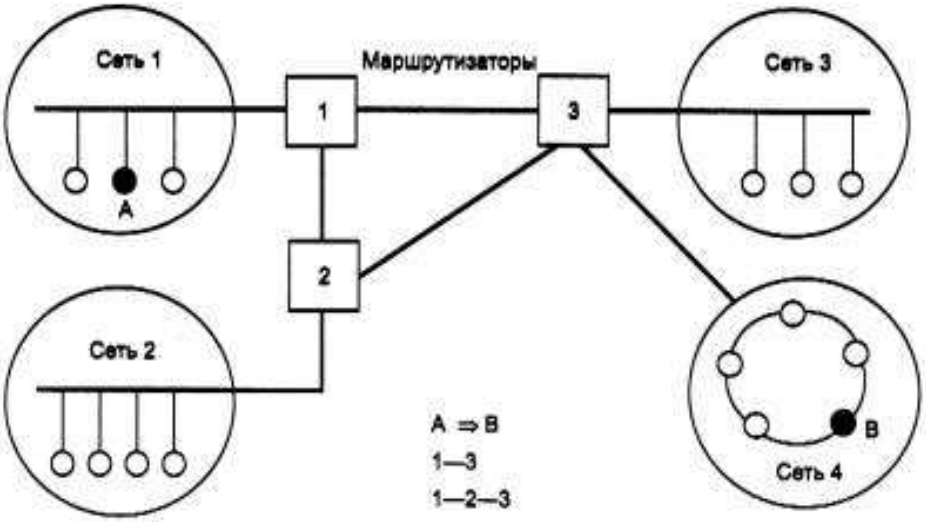


Рис. 2.7. Пример составной сети

Проблема выбора наилучшего пути называется *маршрутизацией*, и ее решение является одной из главных задач сетевого уровня. Эта 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 пытаются приспособиться 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.

В общем случае функции сетевого уровня шире, чем функции передачи сообщений по связям с нестандартной структурой, которые мы сейчас рассмотрели на примере объединения нескольких локальных сетей.

Сетевой уровень решает также задачи согласования разных технологий, упрощения адресации в крупных сетях и создания надежных и гибких барьеров на пути нежелательного трафика между сетями.

Сообщения сетевого уровня принято называть *пакетами* (*packets*). При организации доставки пакетов на сетевом уровне используется 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 и другое, более формальное определение: сеть – это совокупность узлов, сетевой адрес которых содержит один и тот же номер сети.

На сетевом уровне определяются два вида протоколов. Первый вид – *сетевые протоколы* (*routed protocols*) – реализуют продвижение пакетов через сеть. Именно эти протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня. Однако часто к сетевому уровню относят и другой вид протоколов, называемых протоколами обмена маршрутной информацией или просто *протоколами маршрутизации* (*routing protocols*). С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений. Протоколы сетевого уровня реализуются программными модулями операционной системы, а также программными и аппаратными средствами маршрутизаторов.

На сетевом уровне работают протоколы еще одного типа, которые отвечают за отображение адреса узла, используемого на сетевом уровне, в локальный адрес сети. Такие протоколы часто называют *протоколами разрешения адресов – Address Resolution Protocol, ARP*. Иногда их относят не к сетевому уровню, а к канальному, хотя тонкости классификации не изменяют их сути. Примерами протоколов сетевого уровня являются протокол межсетевого взаимодействия IP стека TCP/IP и протокол межсетевого обмена пакетами IPX стека Novell.

Транспортный уровень

На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением. Транспортный уровень (*Transport layer*) обеспечивает приложениям или верхним уровням стека – прикладному и сеансовому – передачу данных с той степенью надежности, которая им требуется. Модель OSI определяет пять классов сервиса, предоставляемых транспортным уровнем. Эти виды сервиса отличаются качеством предоставляемых услуг: срочностью, возможностью восстановления прерван-

ной 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 нижних 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 и циклической нумерации пакетов, установления тайм-аутов доставки и т. п.

Как правило, все протоколы, начиная с транспортного уровня и выше, реализуются программными средствами конечных узлов сети – компонентами их сетевых операционных систем. В качестве примера транспортных протоколов можно привести протоколы TCP и UDP стека TCP/IP и протокол SPX стека Novell. Протоколы нижних четырех уровней обобщенно называют сетевым транспортом или транспортной подсистемой, так как они полностью решают задачу транспортировки сообщений с заданным уровнем качества в составных сетях с произвольной топологией и различными технологиями. Остальные три верхних уровня решают задачи предоставления прикладных сервисов на основании имеющейся транспортной подсистемы.

Сеансовый уровень

Сеансовый уровень (*Session layer*) обеспечивает управление диалогом: фиксирует, какая из сторон является активной в настоящий момент, предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки в длинные передачи, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, а не начинать все с на-

чала. 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*) 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. С помощью средств данного уровня протоколы прикладных уровней могут преодолеть синтаксические различия в представлении данных или же различия в кодах символов, например, кодов ASCII и EBCDIC. На этом уровне 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.

Прикладной уровень

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

Существует очень большое разнообразие служб прикладного уровня. Приведем в качестве примера хотя бы несколько наиболее распространенных реализации файловых служб: NCP в операционной системе Novell NetWare, SMB в Microsoft Windows NT, NFS, FTP и TFTP, входящие в стек TCP/IP.

Сетезависимые и сетенезависимые уровни

Функции всех уровней модели OSI могут быть отнесены к одной из двух групп: либо к функциям, зависящим от конкретной технической реализации сети, либо к функциям, ориентированным на работу с приложениями.

Три нижних уровня – физический, канальный и сетевой – являются сетезависимыми, т. е. протоколы этих уровней тесно связаны с техниче-

ской 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 l00VG-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. На рис. 2.8 показаны уровни модели OSI, на которых работают различные элементы сети.

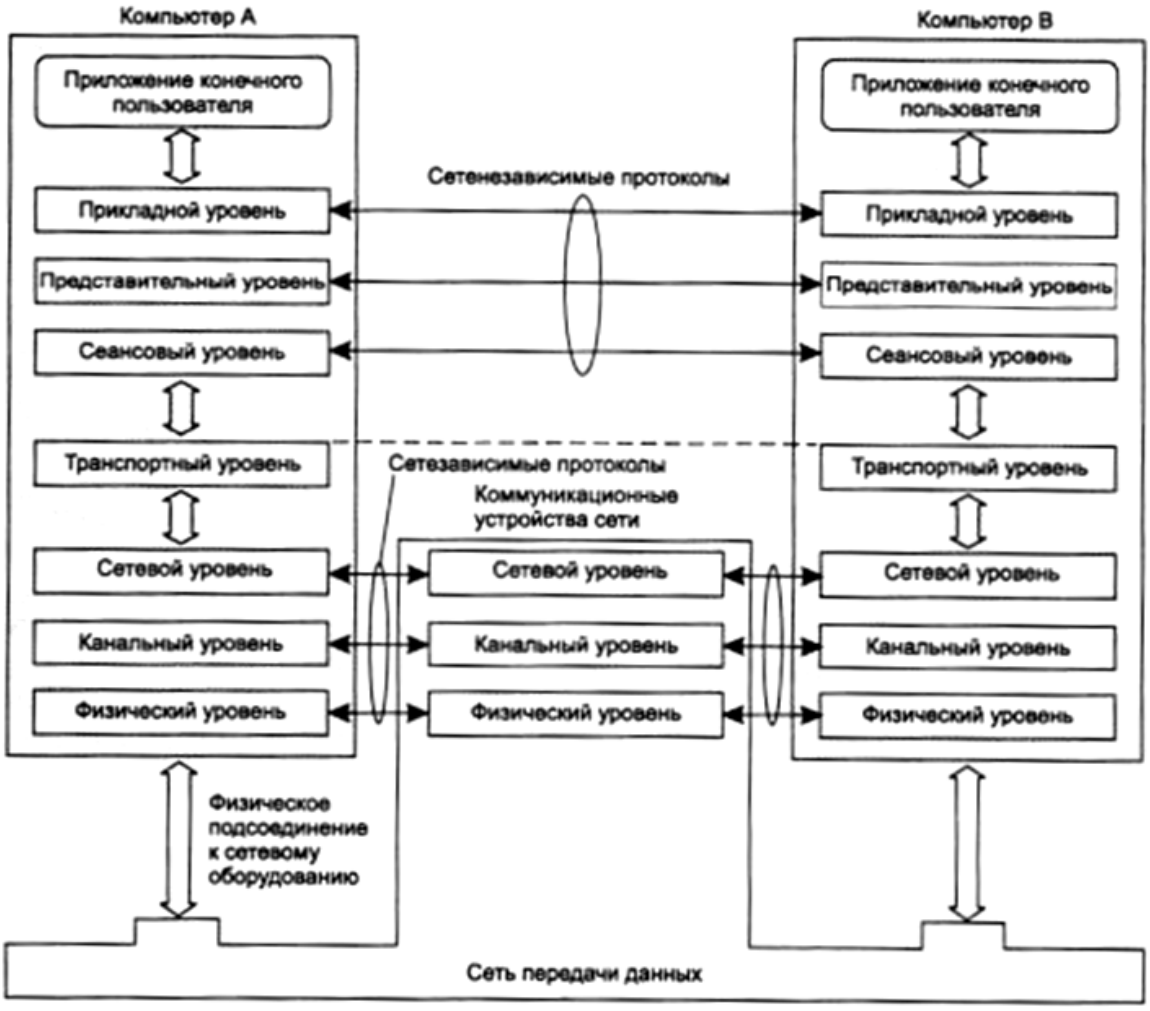


Рис. 2.8. Сетезависимые и сетенезависимые уровни модели OSI

Компьютер с установленной на нем сетевой ОС взаимодействует с другим компьютером с помощью протоколов всех семи уровней. Это взаимодействие компьютеры осуществляют опосредовано через различные коммуникационные устройства – концентраторы, модемы, мосты, коммутаторы, маршрутизаторы, мультиплексоры. В зависимости от типа коммуникационное устройство может работать либо только на физическом уровне (повторитель), либо на физическом и канальном (мост), либо на физическом, канальном и сетевом, иногда захватывая и транспортный уровень (маршрутизатор).

На рис. 2.9 показано соответствие функций различных коммуникационных устройств уровням модели OSI.

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.

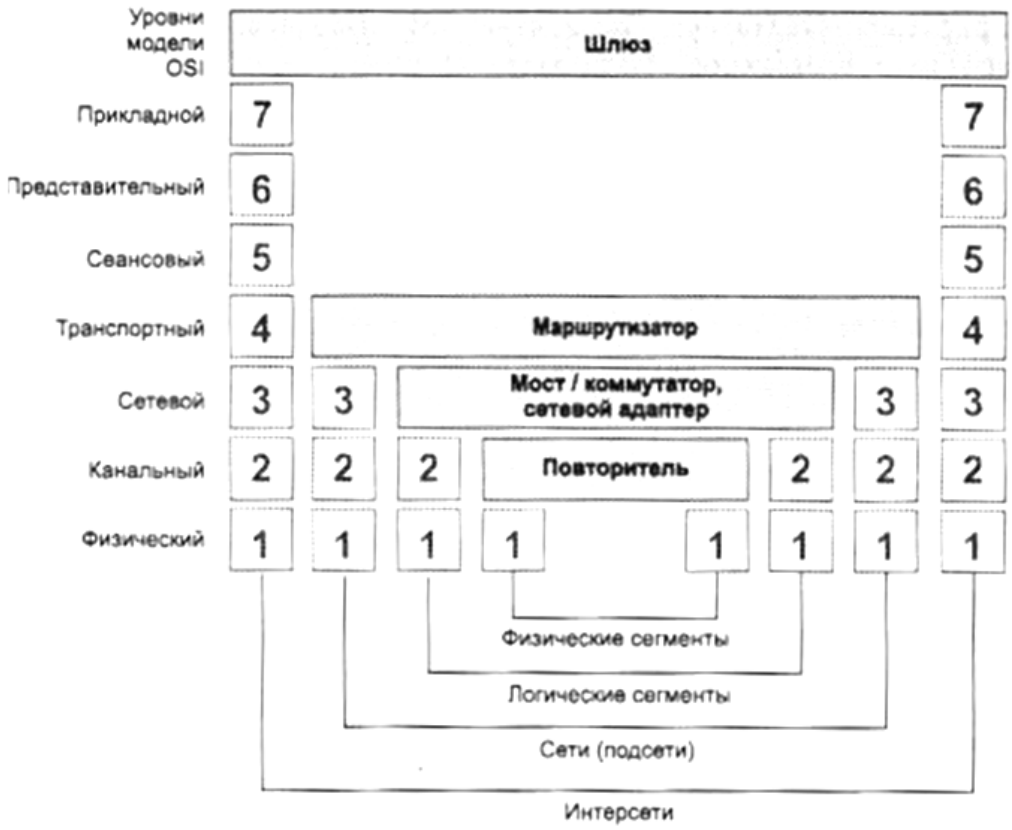


Рис. 2.9. Соответствие функций различных устройств сети уровням модели OSI

2.3. Стандартные стеки коммуникационных протоколов

Важнейшим направлением стандартизации в области вычислительных сетей является стандартизация коммуникационных протоколов. В настоящее время в сетях используется большое количество стеков коммуникационных протоколов. Наиболее популярными являются стеки TCP/IP, IPX/SPX, NetBIOS/SMB, DECnet, SNA и OSI. Все эти стеки, кроме SNA, на нижних уровнях – физическом и канальном – используют одни и те же хорошо стандартизованные протоколы Ethernet, Token Ring, FDDI и некоторые другие, которые позволяют использовать во всех сетях одну и ту же аппаратуру. Зато на верхних уровнях все стеки работают по своим собственным протоколам. Эти протоколы часто не соответствуют рекомендуемому моделью OSI разбиению на уровни. В частности, функции сеансового и представительного уровня, как правило, объединены с прикладным уровнем. Такое несоответствие связано с тем, что модель OSI появилась как результат обобщения уже существующих и реально используемых стеков, а не наоборот.

Стек OSI

Следует четко различать модель OSI и стек OSI. В то время как модель OSI является концептуальной схемой взаимодействия открытых систем, стек OSI представляет собой набор вполне конкретных спецификаций протоколов. В отличие от других стеков протоколов стек OSI полностью соответствует модели OSI, он включает спецификации протоколов для всех семи уровней взаимодействия, определенных в этой модели. На нижних уровнях стек OSI поддерживает Ethernet, Token Ring, FDDI, протоколы глобальных сетей, X.25 и ISDN, т. е. использует разработанные вне стека протоколы нижних уровней, как и все другие стеки. Протоколы сетевого, транспортного и сеансового уровней стека OSI специфицированы и реализованы различными производителями, но распространены пока мало. Наиболее популярными протоколами стека OSI являются прикладные протоколы. К ним относятся протокол передачи файлов FTAM, протокол эмуляции терминала VTP, протоколы справочной службы X.500, электронной почты X.400 и ряд других.

Протоколы стека OSI отличает большая сложность и неоднозначность спецификаций. Эти свойства явились результатом общей политики разработчиков стека, стремившихся учесть в своих протоколах все случаи жизни и все существующие и появляющиеся технологии. К этому нужно еще добавить и последствия большого количества политических компро-

миссов, неизбежных при принятии международных стандартов по такому злободневному вопросу, как построение открытых вычислительных сетей.

Из-за своей сложности протоколы OSI требуют больших затрат вычислительной мощности центрального процессора, что делает их наиболее подходящими для мощных машин, а не для сетей персональных компьютеров.

Стек OSI – международный, независимый от производителей стандарт. Его поддерживает правительство США в своей программе GOSIP, в соответствии с которой все компьютерные сети, устанавливаемые в правительственных учреждениях США после 1990 года, должны или непосредственно поддерживать стек OSI, или обеспечивать средства для перехода на этот стек в будущем. Тем не менее, стек OSI более популярен в Европе, чем в США, так как в Европе осталось меньше старых сетей, работающих по своим собственным протоколам. Большинство организаций пока только планируют переход к стеку OSI, и очень немногие приступили к созданию пилотных проектов. Из тех, кто работает в этом направлении, можно назвать Военно-морское ведомство США и сеть NFSNET. Одним из крупнейших производителей, поддерживающих OSI, является компания AT&T, ее сеть Stargroup полностью базируется на этом стеке.

Стек TCP/IP

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

Стек TCP/IP на нижнем уровне поддерживает все популярные стандарты физического и канального уровней: для локальных сетей это Ethernet, Token Ring, FDDI, для глобальных – протоколы работы на аналоговых коммутируемых и выделенных линиях SLIP, PPP, протоколы территориальных сетей X.25 и ISDN.

Основными протоколами стека, давшими ему название, являются протоколы IP и TCP. Эти протоколы в терминологии модели OSI относятся к сетевому и транспортному уровням соответственно. IP обеспечивает

продвижение пакета по составной сети, а TCP гарантирует надежность его доставки.

За долгие годы использования в сетях различных стран и организаций стек TCP/IP вобрал в себя большое количество протоколов прикладного уровня. К ним относятся такие популярные протоколы, как протокол пересылки файлов FTP, протокол эмуляции терминала Telnet, почтовый протокол SMTP, используемый в электронной почте сети Internet, гипертекстовые сервисы службы WWW и многие другие.

Сегодня стек TCP/IP представляет собой один из самых распространенных стеков транспортных протоколов вычислительных сетей. Действительно, только в сети Internet объединено около 10 миллионов компьютеров по всему миру, которые взаимодействуют друг с другом с помощью стека протоколов TCP/IP.

Стремительный рост популярности Internet привел и к изменениям в расстановке сил в мире коммуникационных протоколов: протоколы TCP/IP, на которых построен Internet, стали быстро теснить бесспорного лидера прошлых лет – стек IPX/SPX компании Novell. Сегодня в мире общее количество компьютеров, на которых установлен стек TCP/IP, сравнялось с общим количеством компьютеров, на которых работает стек IPX/SPX, и это говорит о резком переломе в отношении администраторов локальных сетей к протоколам, используемым на настольных компьютерах, так как именно они составляют подавляющее число мирового компьютерного парка и именно на них раньше почти везде работали протоколы компании Novell, необходимые для доступа к файловым серверам NetWare. Процесс становления стека TCP/IP в качестве стека номер один в любых типах сетей продолжается, и сейчас любая промышленная операционная система обязательно включает программную реализацию этого стека в своем комплекте поставки.

Хотя протоколы TCP/IP неразрывно связаны с Internet и каждый из многомиллионной армады компьютеров Internet работает на основе этого стека, существует большое количество локальных, корпоративных и территориальных сетей, непосредственно не являющихся частями Internet, в которых также используют протоколы TCPДР. Чтобы отличать их от Internet, эти сети называют сетями TCP/IP или просто IP-сетями.

Поскольку стек TCP/IP изначально создавался для глобальной сети Internet, он имеет много особенностей, дающих ему преимущество перед другими протоколами, когда речь заходит о построении сетей, включающих глобальные связи. В частности, очень полезным свойством, делающим возможным применение этого протокола в больших сетях, является его способность фрагментировать пакеты. Действительно, большая со-

ставная сеть часто состоит из сетей, построенных на совершенно разных принципах. В каждой из этих сетей может быть установлена собственная величина максимальной длины единицы передаваемых данных (кадра). В таком случае при переходе из одной сети, имеющей бóльшую максимальную длину, в сеть с меньшей максимальной длиной может возникнуть необходимость деления передаваемого кадра на несколько частей. Протокол IP стека TCP/IP эффективно решает эту задачу.

Другой особенностью технологии TCP/IP является гибкая система адресации, позволяющая более просто по сравнению с другими протоколами аналогичного назначения включать в интерсеть сети других технологий. Это свойство также способствует применению стека TCP/IP для построения больших гетерогенных сетей.

В стеке TCP/IP очень экономно используются возможности широковещательных рассылок. Это свойство совершенно необходимо при работе на медленных каналах связи, характерных для территориальных сетей.

Однако, как и всегда, за получаемые преимущества надо платить, и платой здесь оказываются высокие требования к ресурсам и сложность администрирования IP-сетей. Мощные функциональные возможности протоколов стека TCP/IP требуют для своей реализации высоких вычислительных затрат. Гибкая система адресации и отказ от широковещательных рассылок приводят к наличию в IP-сети различных централизованных служб типа DNS, DHCP и т. п. Каждая из этих служб направлена на облегчение администрирования сети, в том числе и на облегчение конфигурирования оборудования, но в то же время сама требует пристального внимания со стороны администраторов.

Можно приводить и другие доводы за и против стека протоколов Internet, однако факт остается фактом – сегодня это самый популярный стек протоколов, широко используемый как в глобальных, так и локальных сетях.

Стек IPX/SPX

Этот стек является оригинальным стеком протоколов фирмы Novell, разработанным для сетевой операционной системы NetWare еще в начале 80-х годов. Протоколы сетевого и сеансового уровней Internetwork Packet Exchange (IPX) и Sequenced Packet Exchange (SPX), которые дали название стеку, являются прямой адаптацией протоколов XNS фирмы Xerox, распространенных в гораздо меньшей степени, чем стек IPX/SPX. Популярность стека IPX/SPX непосредственно связана с операционной системой Novell NetWare, которая еще сохраняет мировое лидерство по числу уста-

новленных систем, хотя в последнее время ее популярность несколько снизилась и по темпам роста она отстает от Microsoft Windows NT.

Многие особенности стека IPX/SPX обусловлены ориентацией ранних версий ОС NetWare (до версии 4.0) на работу в локальных сетях небольших размеров, состоящих из персональных компьютеров со скромными ресурсами. Понятно, что для таких компьютеров компании Novell нужны были протоколы, на реализацию которых требовалось бы минимальное количество оперативной памяти (ограниченной в IBM-совместимых компьютерах под управлением MS-DOS объемом 640 Кбайт) и которые бы быстро работали на процессорах небольшой вычислительной мощности. В результате протоколы стека IPX/SPX до недавнего времени хорошо работали в локальных сетях и не очень – в больших корпоративных сетях, так как они слишком перегружали медленные глобальные связи широковещательными пакетами, которые интенсивно используются несколькими протоколами этого стека (например, для установления связи между клиентами и серверами). Это обстоятельство, а также тот факт, что стек IPX/SPX является собственностью фирмы Novell и на его реализацию нужно получать лицензию (то есть открытые спецификации не поддерживались), долгое время ограничивали распространенность его только сетями NetWare. Однако с момента выпуска версии NetWare 4.0 Novell внесла и продолжает вносить в свои протоколы серьезные изменения, направленные на их адаптацию для работы в корпоративных сетях. Сейчас стек IPX/ SPX реализован не только в NetWare, но и в нескольких других популярных сетевых ОС, например SCO UNIX, Sun Solaris, Microsoft Windows NT.

Стек NetBIOS/SMB

Этот стек широко используется в продуктах компаний IBM и Microsoft. На физическом и канальном уровнях этого стека используются все наиболее распространенные протоколы Ethernet, Token Ring, FDDI и другие. На верхних уровнях работают протоколы NetBEUI и SMB.

Протокол NetBIOS (Network Basic Input/Output System) появился в 1984 году как сетевое расширение стандартных функций базовой системы ввода/вывода (BIOS) IBM PC для сетевой программы PC Network фирмы IBM. В дальнейшем этот протокол был заменен так называемым протоколом расширенного пользовательского интерфейса NetBEUI – NetBIOS Extended User Interface. Для обеспечения совместимости приложений в качестве интерфейса к протоколу NetBEUI был сохранен интерфейс NetBIOS. Протокол NetBEUI разрабатывался как эффективный протокол, потребляющий немного ресурсов и предназначенный для сетей, насчитывающих не более 200 рабочих станций. Этот протокол содержит много по-

лезных сетевых функций, которые можно отнести к сетевому, транспортному и сеансовому уровням модели OSI, однако с его помощью невозможна маршрутизация пакетов. Это ограничивает применение протокола NetBEUI локальными сетями, не разделенными на подсети, и делает невозможным его использование в составных сетях. Некоторые ограничения NetBEUI снимаются реализацией этого протокола NBF (NetBEUI Frame), которая включена в операционную систему Microsoft Windows NT.

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

Стеки протоколов SNA фирмы IBM, DECnet корпорации Digital Equipment и AppleTalk/AFP фирмы Apple применяются в основном в операционных системах и сетевом оборудовании этих фирм.

На рис. 2.10 показано соответствие некоторых наиболее популярных протоколов уровням модели OSI. Часто это соответствие весьма условно, так как модель OSI – это только руководство к действию, причем достаточно общее, а конкретные протоколы разрабатывались для решения специфических задач, причем многие из них появились до разработки модели OSI.

<table>
<tr><th>Модель OSI</th><th colspan="3">IBM/Microsoft</th><th colspan="3">TCP/IP</th><th colspan="3">Novell</th><th colspan="3">Стек OSI</th></tr>
<tr><td>Прикладной</td><td colspan="3" rowspan="2">SMB</td><td colspan="3" rowspan="2">Telnet,
FTP,
SNMP,
SMTP,
WWW</td><td colspan="3" rowspan="3">NCP,
SAP</td><td colspan="3">X.400
X.500
FTAM</td></tr>
<tr><td>Представительный</td><td colspan="3">Представительный протокол OSI</td></tr>
<tr><td>Сеансовый</td><td colspan="3" rowspan="2">NetBIOS</td><td colspan="3" rowspan="2">TCP</td><td colspan="3">Сеансовый протокол OSI</td></tr>
<tr><td>Транспортный</td><td colspan="3">SPX</td><td colspan="3">Транспортный протокол OSI</td></tr>
<tr><td>Сетевой</td><td colspan="4"></td><td colspan="3">IP,
RIP,
OSPF</td><td colspan="3">IPX,
RIP,
NLSP</td><td colspan="3">ES-ES
IS-IS</td></tr>
<tr><td>Канальный</td><td colspan="10">802.3 (Ethernet), 802.5 (Token Ring), FDDI, Fast Ethernet, SLIP, 100VG-AnyLAN, X.25, ATM, LAP-B, LAP-D, PPP</td></tr>
<tr><td>Физический</td><td colspan="10">Коаксиал, экранированная и неэкранированная витая пара, оптоволокно, радиоволны</td></tr>
</table>

Рис. 2.10. Соответствие популярных стеков протоколов модели OSI

В большинстве случаев разработчики стеков отдавали предпочтение скорости работы сети в ущерб модульности – ни один стек, кроме стека OSI, не разбит на семь уровней. Чаще всего в стеке явно выделяются 3 – 4

уровня: уровень сетевых адаптеров, в котором реализуются протоколы физического и канального уровней, сетевой уровень, транспортный уровень и уровень служб, вбирающий в себя функции сеансового, представительного и прикладного уровней.

2.4. Линии связи. Типы, аппаратура, характеристики линий связи

Линия связи (рис. 2.11) состоит в общем случае из физической среды, по которой передаются электрические информационные сигналы, аппаратуры передачи данных и промежуточной аппаратуры. Синонимом термина *линия связи (line)* является термин *канал связи (channel)*.

Рис. 2.11. Состав линии связи

Типы линий связи

Физическая среда передачи данных (medium) может представлять собой кабель, то есть 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 линии связи разделяются на (рис. 2.12):

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

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 передаются телефонные или телеграфные сигналы, но при отсутствии других возможностей эти линии используются и для передачи компьютерных данных. Скоростные качества и помехозащищенность этих линий оставляют желать много лучшего. Сегодня проводные линии связи быстро вытесняются кабельными.

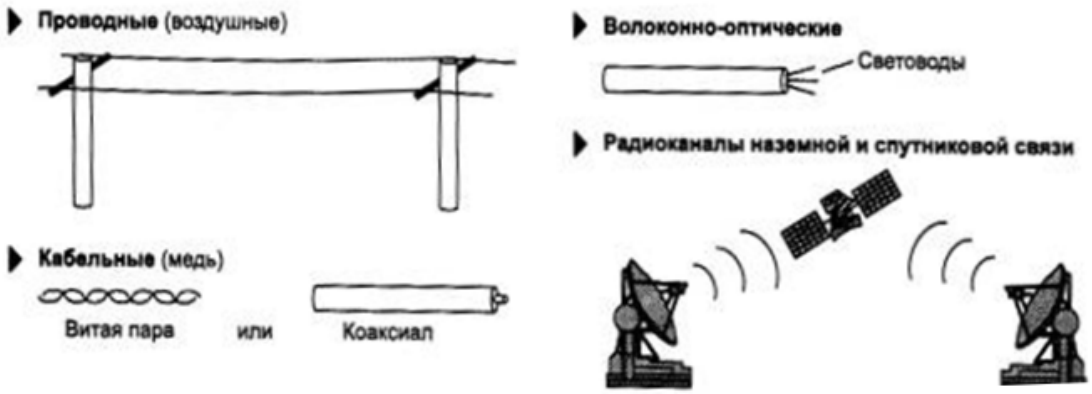


Рис. 2.12. Типы линий связи

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 компьютерных сетях применяются три основных типа кабеля: кабели на основе скрученных пар медных проводов, коаксиальные кабели с медной жилой, а также волоконно-оптические кабели.

Скрученная пара проводов называется *витой парой (twisted pair)*. Витая пара существует в экранированном варианте (Shielded Twistedpair, STP), когда пара медных проводов обертывается в изоляционный экран, и неэкранированном (Unshielded Twistedpair, UTP), когда изоляционная обертка 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 большое количество различных типов радиоканалов, отличающихся как используемым частотным диапазоном, так и дальностью канала. Диапазоны коротких, средних и

длинных волн (КВ, СВ и ДВ), называемые также диапазонами амплитудной модуляции (Amplitude Modulation, AM) 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 (Frequency Modulation, FM), 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 требуется наличие прямой видимости между передатчиком и приемником. Поэтому такие частоты используют либо спутниковые каналы, либо радиорелейные каналы, где это условие выполняется.

В компьютерных сетях сегодня применяются практически все описанные типы физических сред передачи данных, но наиболее перспективными являются волоконно-оптические. На них сегодня строятся 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, а также простотой монтажа. С помощью витой пары обычно подключают конечных абонентов сетей на расстояниях до 100 метров от концентратора. Спутниковые каналы и радиосвязь используются чаще всего в тех случаях, когда кабельные связи применить нельзя, например, при прохождении канала через малонаселенную местность или же для связи с мобильным пользователем сети.

Аппаратура линий связи

Аппаратура передачи данных (АПД или *DCE – Data Circuit terminating Equipment)* непосредственно связывает компьютеры или локальные сети пользователя с линией связи и является, таким образом, пограничным оборудованием. Традиционно аппаратуру передачи данных включают в состав линии связи. Примерами DCE являются модемы, терминальные адаптеры сетей ISDN, оптические модемы, устройства подключения к цифровым каналам. Обычно DCE работает на физическом уровне, отвечая за передачу и прием сигнала нужной формы и мощности в физическую среду.

Аппаратура пользователя линии связи, вырабатывающая данные для передачи по линии связи и подключаемая непосредственно к аппаратуре передачи данных, обобщенно носит название *оконечное оборудование данных (ООД* или *DTE – Data Terminal Equipment)*. Примером DTE могут служить компьютеры или маршрутизаторы локальных сетей. Эту аппаратуру не включают в состав линии связи.

Разделение оборудования на классы DCE и DTE в локальных сетях является достаточно условным. Например, адаптер локальной сети можно считать как принадлежностью компьютера, то есть DTE, так и составной частью канала связи, то есть DCE.

Промежуточная аппаратура обычно используется на линиях связи большой протяженности. Промежуточная аппаратура решает две основные задачи:

- улучшение качества сигнала;
- создание постоянного составного канала связи между двумя абонентами сети.

В локальных сетях промежуточная аппаратура может совсем не использоваться, если протяженность физической среды – кабелей или радиоэфира – позволяет одному сетевому адаптеру принимать сигналы непосредственно от другого сетевого адаптера, без промежуточного усиления. В противном случае применяются устройства типа повторителей и концентраторов.

В глобальных сетях необходимо обеспечить качественную передачу сигналов на расстояния в сотни и тысячи километров. Поэтому без усилителей сигналов, установленных через определенные расстояния, построить территориальную линию связи невозможно. В глобальной сети необходима также и промежуточная аппаратура другого рода – мультиплексоры, демультиплексоры и коммутаторы. Эта аппаратура решает вторую указанную задачу, т. е. создает между двумя абонентами сети составной канал из некоммутируемых отрезков физической среды – кабелей с усилителями. Важно отметить, что приведенные на рис. 2.11 мультиплексоры, демультиплексоры и коммутаторы образуют составной канал на долговременной основе, например, на месяц или год, причем абонент не может влиять на процесс коммутации этого канала – эти устройства управляются по отдельным входам, абоненту недоступным (на рисунке не показаны). Наличие промежуточной коммутационной аппаратуры избавляет создателей глобальной сети от необходимости прокладывать отдельную кабельную линию для каждой пары соединяемых узлов сети. Вместо этого между мультиплексорами и коммутаторами используется высокоскоростная физическая среда, например волоконно-оптический или коаксиальный кабель, по которому передаются одновременно данные от большого числа сравнительно низкоскоростных абонентских линий. А когда нужно образовать постоянное соединение между какими-либо двумя конечными узлами сети, находящимися, например, в разных городах, то мультиплексо-

ры, коммутаторы и демультиплексоры настраиваются оператором канала соответствующим образом. Высокоскоростной канал обычно называют уплотненным каналом.

Промежуточная аппаратура канала связи прозрачна для пользователя, он ее не замечает и не учитывает в своей работе. Для него важны только качество полученного канала, влияющее на скорость передачи дискретных данных. В действительности же промежуточная аппаратура образует сложную сеть, которую называют первичной сетью, так как сама по себе она никаких высокоуровневых служб (например, файловой или передачи голоса) не поддерживает, а только служит основой для построения компьютерных, телефонных или иных сетей.

В зависимости от типа промежуточной аппаратуры все линии связи делятся на *аналоговые* и *цифровые*. В *аналоговых* линиях промежуточная аппаратура предназначена для усиления аналоговых сигналов, т. е. сигналов, которые имеют непрерывный диапазон значений. Такие линии связи традиционно применялись в телефонных сетях для связи АТС между собой. Для создания высокоскоростных каналов, которые мультиплексируют несколько низкоскоростных аналоговых абонентских каналов, при аналоговом подходе обычно используется техника частотного мультиплексирования (Frequency Division Multiplexing, FDM).

В *цифровых* линиях связи передаваемые сигналы имеют конечное число состояний. Как правило, элементарный сигнал, т. е. сигнал, передаваемый за один такт работы передающей аппаратуры, имеет два или три состояния, которые передаются в линиях связи импульсами прямоугольной формы. С помощью таких сигналов передаются как компьютерные данные, так и оцифрованные речь и изображение. В цифровых каналах связи используется промежуточная аппаратура, которая улучшает форму импульсов и обеспечивает их ресинхронизацию, т. е. восстанавливает период их следования. Промежуточная аппаратура образования высокоскоростных цифровых каналов (мультиплексоры, демультиплексоры, коммутаторы) работает по принципу временно́го мультиплексирования каналов (Time Division Multiplexing, TDM), когда каждому низкоскоростному каналу выделяется определенная доля времени (тайм-слот или квант) высокоскоростного канала.

Аппаратура передачи дискретных компьютерных данных по аналоговым и цифровым линиям связи существенно отличается, т. к. в первом случае линия связи предназначена для передачи сигналов произвольной формы и не предъявляет никаких требований к способу представления единиц и нулей аппаратурой передачи данных, а во втором все параметры

передаваемых линией импульсов стандартизованы. Другими словами, на цифровых линиях связи протокол физического уровня определен, а на аналоговых линиях – нет.

Характеристики линий связи

К основным характеристикам линий связи относятся:

- амплитудно-частотная характеристика;
- полоса пропускания;
- затухание;
- помехоустойчивость;
- перекрестные наводки на ближнем конце линии;
- пропускная способность;
- достоверность передачи данных;
- удельная стоимость.

В первую очередь разработчика вычислительной сети интересуют пропускная способность и достоверность передачи данных, поскольку эти характеристики прямо влияют на производительность и надежность создаваемой сети. Пропускная способность и достоверность – это характеристики как линии связи, так и способа передачи данных. Поэтому если способ передачи (протокол) уже определен, то известны и эти характеристики. Например, пропускная способность цифровой линии всегда известна, т. к. на ней 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 линии связи до того, как для нее определен протокол физического уровня. Именно в таких случаях, когда только предстоит определить, какой из множества существующих протоколов можно использовать на данной линии, очень важными являются остальные характеристики линии, такие как полоса пропускания, перекрестные наводки, помехоустойчивость и другие характеристики.

Для определения характеристик линии связи часто используют анализ ее 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 используются 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.

2.5. Стандарты кабелей

Кабель – это достаточно сложное изделие, состоящее из проводников, слоев экрана и изоляции. В некоторых случаях в состав кабеля входят разъемы, с помощью которых кабели присоединяются к оборудованию. Кроме этого, для обеспечения быстрой перекоммутации кабелей и оборудования используются различные электромеханические устройства, называемые кроссовыми секциями, кроссовыми коробками или шкафами.

В компьютерных сетях применяются кабели, удовлетворяющие определенным стандартам, что позволяет строить кабельную систему сети из кабелей и соединительных устройств разных производителей. Сегодня наиболее употребительными стандартами в мировой практике являются следующие.

Американский стандарт EIA/TIA-568A, который был разработан совместными усилиями нескольких организаций – ANSI, EIA/TIA и лабораторией Underwriters Labs (UL). Стандарт EIA/TIA-568 разработан на основе предыдущей версии стандарта EIA/TIA-568 и дополнений к этому стандарту TSB-36 и TSB-40A.

Международный стандарт ISO/IEC 11801.

Европейский стандарт EN50173.

Эти стандарты близки между собой и по многим позициям предъявляют к кабелям идентичные требования. Однако есть и различия между этими стандартами, например, в международный стандарт 11801 и европейский EN50173 вошли некоторые типы кабелей, которые отсутствуют в стандарте EIA/TAI-568A.

До появления стандарта EIA/TIA большую роль играл американский стандарт *системы категорий кабелей* Underwriters Labs, разработанный совместно с компанией Anixter. Позже этот стандарт вошел в стандарт EIA/TIA-568.

Кроме этих открытых стандартов, многие компании в свое время разработали свои фирменные стандарты, из которых до сих пор имеет практическое значение только один – стандарт компании IBM.

В стандартах кабелей оговаривается достаточно много характеристик, из которых наиболее важные перечислены ниже.

Затухание (Attenuation). Затухание измеряется в децибелах на метр для определенной частоты или диапазона частот сигнала.

Перекрестные наводки на ближнем конце (Near End Cross Talk, NEXT). Измеряются в децибелах для определенной частоты сигнала.

Импеданс (волновое сопротивление) – это полное (активное и реактивное) сопротивление в электрической цепи. Импеданс измеряется в омах и является относительно постоянной величиной для кабельных систем (например, для коаксиальных кабелей, используемых в стандартах Ethernet, импеданс кабеля должен составлять 50 Ом). Для неэкранированной витой пары наиболее часто используемые значения импеданса – 100 и 120 Ом. В области высоких частот (100 – 200 МГц) импеданс зависит от частоты.

Активное сопротивление – это сопротивление постоянному току в электрической цепи. В отличие от импеданса активное сопротивление не зависит от частоты и возрастает с увеличением длины кабеля.

Емкость – это свойство металлических проводников накапливать энергию. Два электрических проводника в кабеле, разделенные диэлектриком, представляют собой конденсатор, способный накапливать заряд. Емкость является нежелательной величиной, поэтому следует стремиться к тому, чтобы она была как можно меньше (иногда применяют термин «паразитная емкость»). Высокое значение емкости в кабеле приводит к искажению сигнала и ограничивает полосу пропускания линии.

Уровень внешнего электромагнитного излучения или электрический шум. Электрический шум – это нежелательное переменное напряжение в проводнике. Электрический шум бывает двух типов – фоновый и импульсный. Электрический шум можно также разделить на низко-, средне- и высокочастотный. Источниками фонового электрического шума в диапазоне до 150 кГц являются линии электропередачи, телефоны и лампы дневного света; в диапазоне от 150 кГц до 20 МГц – компьютеры, принтеры, ксероксы; в диапазоне от 20 МГц до 1 ГГц – телевизионные и радиопередатчики, микроволновые печи. Основными источниками импульсного электрического шума являются моторы, переключатели и сварочные агрегаты. Электрический шум измеряется в милливольтах.

Диаметр или *площадь сечения* проводника. Для медных проводников достаточно употребительной является американская система AWG (American Wire Gauge), которая вводит некоторые условные типы проводников, например 22 AWG, 24 AWG, 26 AWG. Чем больше номер типа проводника, тем меньше его диаметр. В вычислительных сетях наиболее употребительными являются типы проводников, приведенные выше в качестве примеров. В европейских и международных стандартах диаметр проводника указывается в миллиметрах. Естественно, приведенный перечень характеристик далеко не полон, причем в нем представлены только электромагнитные характеристики и его нужно дополнить механическими и конструктивными характеристиками, определяющими тип изоляции,

конструкцию разъема и т. п. Помимо универсальных характеристик, таких, например, как затухание, которые применимы для всех типов кабелей, существуют характеристики, которые применимы только к определенному типу кабеля. Например, параметр *шаг скрутки проводов* используется только для характеристики витой пары, а параметр *NEXT* применим только к многопарным кабелям на основе витой пары.

Основное внимание в современных стандартах уделяется кабелям на основе витой пары и волоконно-оптическим кабелям.

Кабели на основе неэкранированной витой пары

Медный неэкранированный кабель UTP в зависимости от электрических и механических характеристик разделяется на 5 категорий (Category 1 – Category 5). Кабели категорий 1 и 2 были определены в стандарте EIA/TIA-568, но в стандарт 568A уже не вошли, как устаревшие.

Кабели *категории 1* применяются там, где требования к скорости передачи минимальны. Обычно это кабель для цифровой и аналоговой передачи голоса и низкоскоростной (до 20 Кбит/с) передачи данных. До 1983 года это был основной тип кабеля для телефонной разводки.

Кабели *категории 2* были впервые применены фирмой IBM при построении собственной кабельной системы. Главное требование к кабелям этой категории – способность передавать сигналы со спектром до 1 МГц.

Кабели *категории 3* были стандартизованы в 1991 году, когда был разработан Стандарт телекоммуникационных кабельных систем для коммерческих зданий (EIA-568), на основе которого затем был создан действующий стандарт EIA-568A. Стандарт EIA-568 определил электрические характеристики кабелей категории 3 для частот в диапазоне до 16 МГц, поддерживающих, таким образом, высокоскоростные сетевые приложения. Кабель категории 3 предназначен как для передачи данных, так и для передачи голоса. Шаг скрутки проводов равен примерно 3 витка на 1 фут (30,5 см). Кабели категории 3 сейчас составляют основу многих кабельных систем зданий, в которых они используются для передачи и голоса, и данных.

Кабели *категории 4* представляют собой несколько улучшенный вариант кабелей категории 3. Кабели категории 4 обязаны выдерживать тесты на частоте передачи сигнала 20 МГц и обеспечивать повышенную помехоустойчивость и низкие потери сигнала. Кабели категории 4 хорошо подходят для применения в системах с увеличенными расстояниями (до 135 метров) и в сетях Token Ring с пропускной способностью 16 Мбит/с. На практике используются редко.

Кабели *категории 5* были специально разработаны для поддержки высокоскоростных протоколов. Поэтому их характеристики определяются в диапазоне до 100 МГц. Большинство новых высокоскоростных стандартов ориентируются на использование витой пары 5 категории. На этом кабеле работают протоколы со скоростью передачи данных 100 Мбит/с – FDDI (с физическим стандартом TP-PMD), Fast Ethernet, l00VG-AnyLAN, а также более скоростные протоколы – ATM на скорости 155 Мбит/с и Gigabit Ethernet на скорости 1000 Мбит/с (вариант Gigabit Ethernet на витой паре категории 5 стал стандартом в июне 1999 года). Кабель категории 5 пришел на замену кабелю категории 3, и сегодня все новые кабельные системы крупных зданий строятся именно на этом типе кабеля (в сочетании с волоконно-оптическим).

Наиболее важные электромагнитные характеристики кабеля категории 5 имеют следующие значения:

- полное волновое сопротивление в диапазоне частот до 100 МГц равно 100 Ом (стандарт ISO 11801 допускает также кабель с волновым сопротивлением 120 Ом);
- величина перекрестных наводок NEXT в зависимости от частоты сигнала должна принимать значения не менее 74 дБ на частоте 150 кГц и не менее 32 дБ на частоте 100 МГц;
- затухание имеет предельные значения от 0,8 дБ (на частоте 64 кГц) до 22 дБ (на частоте 100 МГц);
- активное сопротивление не должно превышать 9,4 Ом на 100 м;
- емкость кабеля не должна превышать 5,6 нф на 100 м.

Все кабели UTP независимо от их категории выпускаются в 4-парном исполнении. Каждая из четырех пар кабеля имеет определенный цвет и шаг скрутки. Обычно две пары предназначены для передачи данных, а две – для передачи голоса.

Для соединения кабелей с оборудованием используются вилки и розетки RJ-45, представляющие 8-контактные разъемы, похожие на обычные телефонные разъемы RJ-11.

Особое место занимают кабели категорий 6 и 7, которые промышленность начала выпускать сравнительно недавно. Для кабеля категории 6 характеристики определяются до частоты 200 МГц, а для кабелей категории 7 – до 600 МГц. Кабели категории 7 обязательно экранируются, причем как каждая пара, так и весь кабель в целом. Кабель категории 6 может быть как экранированным, так и неэкранированным. Основное назначение этих кабелей – поддержка высокоскоростных протоколов на отрезках кабеля большей длины, чем кабель UTP категории 5. Некоторые специали-

сты сомневаются в необходимости применения кабелей категории 7, так как стоимость кабельной системы при их использовании получается соизмеримой по стоимости сети с использованием волоконно-оптических кабелей, а характеристики кабелей на основе оптических волокон выше.

Кабели на основе экранированной витой пары

Экранированная витая пара STP хорошо защищает передаваемые сигналы от внешних помех, а также меньше излучает электромагнитных колебаний вовне, что защищает, в свою очередь, пользователей сетей от вредного для здоровья излучения. Наличие заземляемого экрана удорожает кабель и усложняет его прокладку, так как требует выполнения качественного заземления. Экранированный кабель применяется только для передачи данных, а голос по нему не передают.

Основным стандартом, определяющим параметры экранированной витой пары, является фирменный стандарт IBM. В этом стандарте кабели делятся не на категории, а на типы – Type I, Type 2,..., Type 9.

Основным типом экранированного кабеля является кабель Type 1 стандарта IBM. Он состоит из двух пар скрученных проводов, экранированных проводящей оплеткой, которая заземляется. Электрические параметры кабеля Type 1 примерно соответствуют параметрам кабеля UTP категории 5. Однако волновое сопротивление кабеля Type 1 равно 150 Ом (UTP категории 5 имеет волновое сопротивление 100 Ом), поэтому простое «улучшение» кабельной проводки сети путем замены неэкранированной пары UTP на STP Type 1 невозможно. Трансиверы, рассчитанные на работу с кабелем, имеющим волновое сопротивление 100 Ом, будут плохо работать на волновое сопротивление 150 Ом. Поэтому при использовании STP Type 1 необходимы соответствующие трансиверы. Такие трансиверы имеются в сетевых адаптерах Token Ring, т. к. эти сети разрабатывались для работы на экранированной витой паре. Некоторые другие стандарты также поддерживают кабель STP Type I, например l00VG-AnyLAN, а также Fast Ethernet (хотя основным типом кабеля для Fast Ethernet является UTP категории 5). В случае если технология может использовать UTP и STP, нужно уточнить, на какой тип кабеля рассчитаны приобретаемые трансиверы. Сегодня кабель STP Type 1 включен в стандарты EIA/TIA-568A, ISO 11801 и EN50173, т. е. приобрел международный статус.

Экранированные витые пары используются также в кабеле IBM Type 2, который представляет кабель Type 1 с добавленными двумя парами неэкранированного провода для передачи голоса.

Для присоединения экранированных кабелей к оборудованию используются разъемы конструкции IBM.

Не все типы кабелей стандарта IBM относятся к экранированным кабелям, некоторые определяют характеристики неэкранированного телефонного кабеля (Type 3) и оптоволоконного кабеля (Type 5).

Коаксиальные кабели

Существует большое количество типов коаксиальных кабелей, используемых в сетях различного типа – телефонных, телевизионных и компьютерных. Ниже приводятся основные типы и характеристики этих кабелей.

RG-8 и RG-11 – «толстый» коаксиальный кабель, разработанный для сетей Ethernet 10Base-5. Имеет волновое сопротивление 50 Ом и внешний диаметр 0,5 дюйма (около 12 мм). Этот кабель имеет достаточно толстый внутренний проводник диаметром 2,17 мм, который обеспечивает хорошие механические и электрические характеристики (затухание на частоте 10 МГц – не хуже 18 дБ/км). Зато этот кабель сложно монтировать – он плохо гнется.

RG-58/U, RG-58 A/U и RG-58 C/U – разновидности «тонкого» коаксиального кабеля для сетей Ethernet l0Base-2. Кабель RG-58/U имеет сплошной внутренний проводник, а кабель RG-58 A/U – многожильный. Кабель RG-58 C/U проходит «военную приемку». Все эти разновидности кабеля имеют волновое сопротивление 50 Ом, но обладают худшими механическими и электрическими характеристиками по сравнению с «толстым» коаксиальным кабелем. Тонкий внутренний проводник 0,89 мм не так прочен, зато обладает гораздо большей гибкостью, удобной при монтаже. Затухание в этом типе кабеля выше, чем в «толстом» коаксиальном кабеле, что приводит к необходимости уменьшать длину кабеля для получения одинакового затухания в сегменте. Для соединения кабелей с оборудованием используется разъем типа BNC.

RG-59 – телевизионный кабель с волновым сопротивлением 75 Ом. Широко применяется в кабельном телевидении.

RG-62 – кабель с волновым сопротивлением 93 Ома, использовался в сетях ArcNet, оборудование которых сегодня практически не выпускается. Коаксиальные кабели с волновым сопротивлением 50 Ом (т. е. «тонкий» и «толстый») описаны в стандарте EIA/TIA-568. Новый стандарт EIA/TIA-568A коаксиальные кабели не описывает, как морально устаревшие.

Волоконно-оптические кабели

Волоконно-оптические кабели состоят из центрального проводника света (сердцевины) – стеклянного волокна, окруженного другим слоем стекла – оболочкой, обладающей меньшим показателем преломления, чем сердцевина. Распространяясь по сердцевине, лучи света не выходят за ее пределы, отражаясь от покрывающего слоя оболочки. В зависимости от распределения показателя преломления и от величины диаметра сердечника различают:

- многомодовое волокно со ступенчатым изменением показателя преломления (рис. 2.13, *а*);
- многомодовое волокно с плавным изменением показателя преломления (см. рис. 2.13, *б*);
- одномодовое волокно (см. рис. 2.13, *в*).

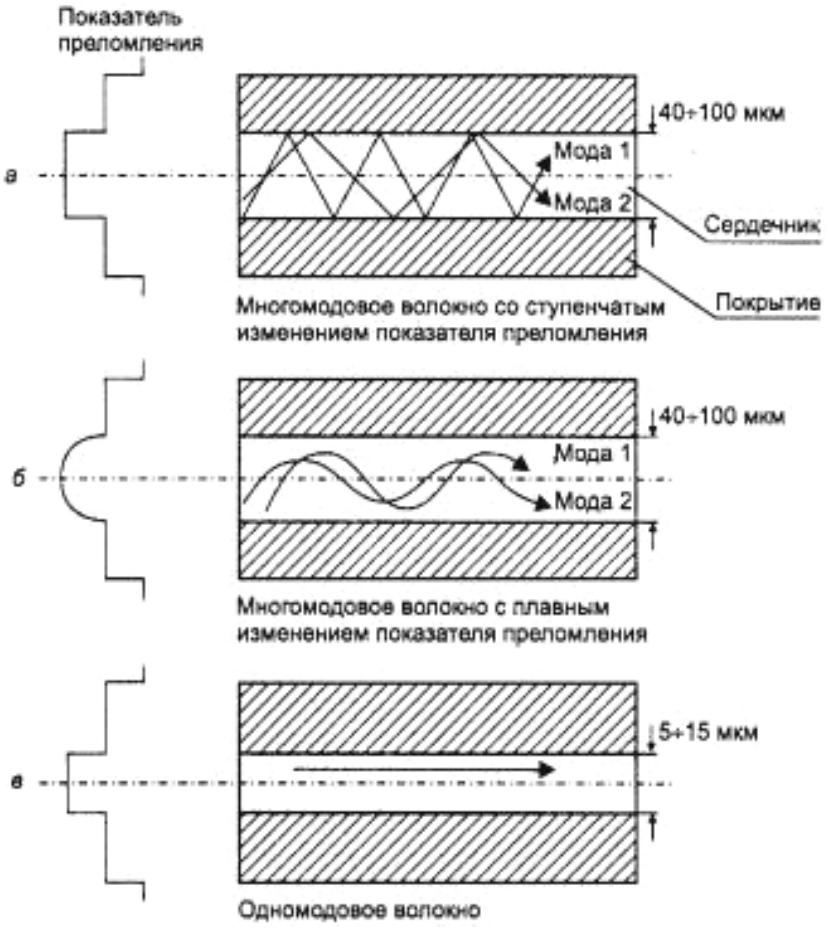


Рис. 2.13. Типы оптического кабеля

Понятие «мода» описывает режим распространения световых лучей во внутреннем сердечнике кабеля. В *одномодовом кабеле (Single Mode Fiber, SMF)* используется центральный проводник очень малого диаметра,

соизмеримого с длиной волны света – от 5 до 10 мкм. При этом практически все лучи света распространяются вдоль оптической оси световода, не отражаясь от внешнего проводника. Полоса пропускания одномодового кабеля очень широкая – до сотен гигагерц на километр. Изготовление тонких качественных волокон для одномодового кабеля представляет сложный технологический процесс, что делает одномодовый кабель достаточно дорогим. Кроме того, в волокно такого маленького диаметра достаточно сложно направить пучок света, не потеряв при этом значительную часть его энергии.

В *многомодовых кабелях (Multi Mode Fiber, MMF)* используются более широкие внутренние сердечники, которые легче изготовить технологически. В стандартах определены два наиболее употребительных многомодовых кабеля – 62,5/125 мкм и 50/125 мкм, где 62,5 мкм или 50 мкм – это диаметр центрального проводника, а 125 мкм – диаметр внешнего проводника.

В многомодовых кабелях во внутреннем проводнике одновременно существуют несколько световых лучей, отражающихся от внешнего проводника под разными углами. Угол отражения луча называется модой луча. В многомодовых кабелях с плавным изменением коэффициента преломления режим распространения каждой моды имеет более сложный характер.

Многомодовые кабели имеют более узкую полосу пропускания – от 500 до 800 МГц/км. Сужение полосы происходит из-за потерь световой энергии при отражениях, а также из-за интерференции лучей разных мод.

В качестве источников излучения света в волоконно-оптических кабелях применяются:

- светодиоды;
- полупроводниковые лазеры.

Для одномодовых кабелей применяются только полупроводниковые лазеры, так как при таком малом диаметре оптического волокна световой поток, создаваемый светодиодом, невозможно без больших потерь направить в волокно. Для многомодовых кабелей используются более дешевые светодиодные излучатели.

Для передачи информации применяется свет с длиной волны 1550 нм (1,55 мкм), 1300 нм (1,3 мкм) и 850 нм (0,85 мкм). Светодиоды могут излучать свет с длиной волны 850 нм и 1300 нм. Излучатели с длиной волны 850 нм существенно дешевле, чем излучатели с длиной волны 1300 нм, но полоса пропускания кабеля для волн 850 нм у́же, например, 200 МГц/км вместо 500 МГц/км.

Лазерные излучатели работают на длинах волн 1300 и 1550 нм. Быстродействие современных лазеров позволяет модулировать световой поток с частотами 10 ГГц и выше. Лазерные излучатели создают когерентный поток света, за счет чего потери в оптических волокнах становятся меньше, чем при использовании некогерентного потока светодиодов.

Использование только нескольких длин волн для передачи информации в оптических волокнах связанно с особенностью их амплитудно-частотной характеристики. Именно для этих дискретных длин волн наблюдаются ярко выраженные максимумы передачи мощности сигнала, а для других волн затухание в волокнах существенно выше.

Волоконно-оптические кабели присоединяют к оборудованию разъемами MIC, ST и SC.

Волоконно-оптические кабели обладают отличными характеристиками всех типов – электромагнитными, механическими (хорошо гнутся, а в соответствующей изоляции обладают хорошей механической прочностью). Однако у них есть один серьезный недостаток – сложность соединения волокон с разъемами и между собой при необходимости наращивания длины кабеля.

Сама стоимость волоконно-оптических кабелей ненамного превышает стоимость кабелей на витой паре, однако проведение монтажных работ с оптоволокном обходится намного дороже из-за трудоемкости операций и высокой стоимости применяемого монтажного оборудования. Так, присоединение оптического волокна к разъему требует проведения высокоточной обрезки волокна в плоскости, строго перпендикулярной к оси волокна, а также выполнения соединения путем сложной операции склеивания, а не обжатия, как это делается для витой пары. Выполнение же некачественных соединений сразу резко сужает полосу пропускания волоконно-оптических кабелей и линий.

2.6. Методы передачи дискретных данных на физическом уровне

При передаче дискретных данных по каналам связи применяются два основных типа физического 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.

Аналоговая модуляция

Аналоговая модуляция применяется для передачи дискретных данных по каналам с узкой полосой частот, типичным представителем которых является канал тональной частоты, предоставляемый в распоряжение пользователям общественных телефонных сетей. Типичная амплитудно-частотная характеристика канала тональной частоты представлена на рис. 2.14. Этот канал передает частоты в диапазоне от 300 до 3400 Гц, таким образом, его полоса пропускания равна 3100 Гц. Хотя человеческий голос имеет гораздо более широкий спектр – примерно от 100 Гц до 10 кГц – для приемлемого качества передачи речи диапазон в 3100 Гц является хорошим решением. Строгое ограничение полосы пропускания тонального канала связано с использованием аппаратуры уплотнения и коммутации каналов в телефонных сетях.

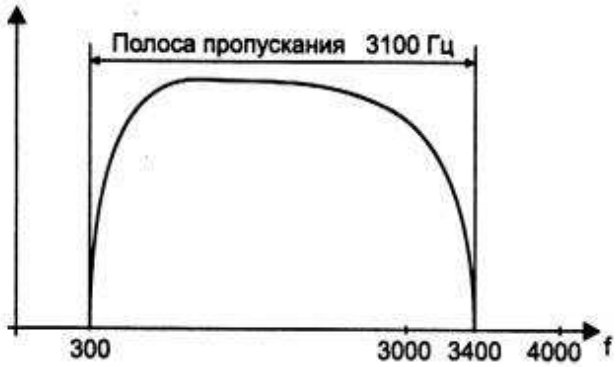


Рис. 2.14. Амплитудно-частотная характеристика канала тональной частоты

Устройство, которое выполняет функции модуляции несущей синусоиды на передающей стороне и демодуляции на приемной стороне, носит название *модем (модулятор – демодулятор)*.

Методы аналоговой модуляции

Аналоговая модуляция является таким способом физического кодирования, при котором информация кодируется изменением амплитуды, частоты или фазы синусоидального сигнала несущей частоты. Основные способы аналоговой модуляции показаны на рис. 2.15. На диаграмме (см. рис. 2.15, *а*) показана последовательность бит исходной информации, представленная потенциалами высокого уровня для логической единицы и потенциалом нулевого уровня для логического нуля. Такой способ кодирования называется потенциальным кодом, который часто используется при передаче данных между блоками компьютера.

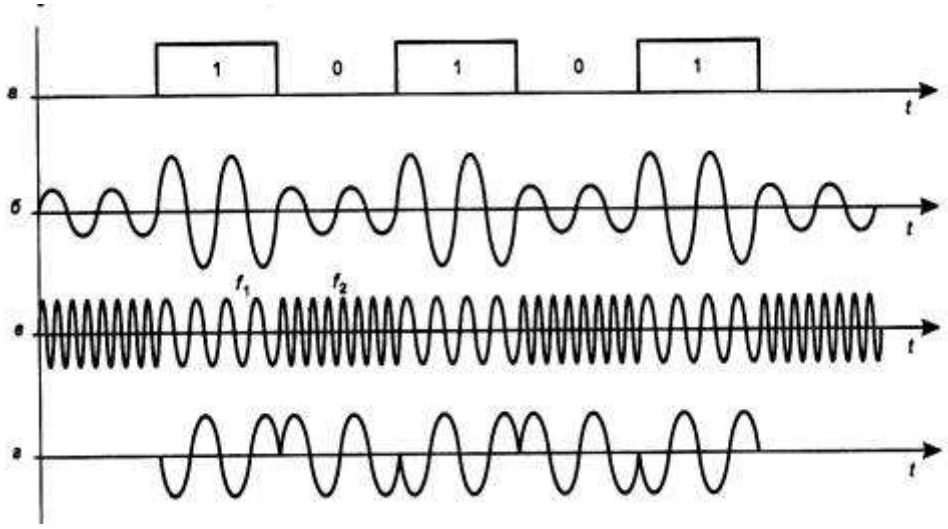


Рис. 2.15. Различные типы модуляции

При *амплитудной модуляции* (см. рис. 2.15, *б*) для логической единицы выбирается один уровень амплитуды синусоиды несущей частоты, а для логического нуля – другой. Этот способ редко используется в чистом виде на практике из-за низкой помехоустойчивости, но часто применяется в сочетании с другим видом модуляции – фазовой модуляцией.

При *частотной модуляции* (см. рис. 2.15, *в*) значения 0 и 1 исходных данных передаются синусоидами с различной частотой – f0 и f1. Этот способ модуляции не требует сложных схем в модемах и обычно применяется в низкоскоростных модемах, работающих на скоростях 300 или 1200 бит/с.

При *фазовой модуляции* (см. рис. 2.15, *г*) значениям данных 0 и 1 соответствуют сигналы одинаковой частоты, но с различной фазой, например 0 и 180 градусов или 0, 90, 180 и 270 градусов.

В скоростных модемах часто используются комбинированные методы модуляции, как правило, амплитудная в сочетании с фазовой.

Цифровое кодирование

При цифровом кодировании дискретной информации применяют потенциальные и импульсные коды.

В потенциальных кодах для представления логических единиц и нулей используется только значение потенциала сигнала, а его перепады, формирующие законченные импульсы, во внимание не принимаются. Импульсные коды позволяют представить двоичные данные либо импульсами определенной полярности, либо частью импульса – перепадом потенциала определенного направления.

Требования к методам цифрового кодирования

При использовании прямоугольных импульсов для передачи дискретной информации необходимо выбрать такой способ кодирования, который одновременно достигал бы нескольких целей:

- имел при одной и той же битовой скорости наименьшую ширину спектра результирующего сигнала;
- обеспечивал синхронизацию между передатчиком и приемником;
- обладал способностью распознавать ошибки;
- обладал низкой стоимостью реализации.

Более узкий спектр сигналов позволяет на одной и той же линии (с одной и той же полосой пропускания) добиваться более высокой скорости 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.

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 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. 2.16), так что информация снимается с линии данных только в момент прихода тактового импульса. В сетях использование этой схемы вызывает трудности из-за неоднородности характеристик проводников в кабелях. На больших расстояниях неравномерность скорости распространения сигнала может привести к тому, что тактовый импульс придет настолько позже или раньше соответствующего сигнала данных, что бит данных будет пропущен или считан повторно. Другой причиной, по которой в сетях отказываются от использования тактирующих импульсов, является экономия проводников в дорогостоящих кабелях.

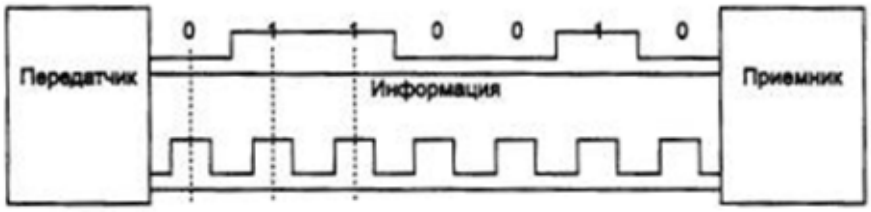


Рис. 2.16. Синхронизация приемника и передатчика на небольших расстояниях

Поэтому в сетях применяются так называемые самосинхронизирующиеся коды, сигналы которых несут для передатчика указания о том, в какой момент времени нужно осуществлять распознавание очередного бита (или нескольких бит, если код ориентирован более чем на два состояния сигнала). Любой резкий перепад сигнала (так называемый фронт) может служить хорошим указанием для синхронизации приемника с передатчиком.

Распознавание и коррекцию искаженных данных сложно осуществить средствами физического уровня, поэтому чаще всего эту работу берут на себя протоколы, лежащие выше, – канальный, сетевой, транспортный или прикладной. 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.

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

Потенциальный код без возвращения к нулю

На рис. 2.17, *а* показан упомянутый ранее метод потенциального кодирования, называемый также кодированием без возвращения к нулю (Non Return to Zero, NRZ). Последнее название отражает то обстоятельство, что 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, обладает хорошей распознаваемостью ошибок (из-за двух резко отличающихся потенциалов), но не обладает свойством самосинхронизации. При передаче длинной последовательности единиц или нулей сигнал на линии не изменяется, поэтому приемник лишен возможности определять по входному сигналу моменты времени, когда нужно в очередной раз считывать данные.

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 результате в чистом виде код NRZ в сетях не используется. Тем не менее,

используются его различные модификации, в которых устраняют как плохую самосинхронизацию кода NRZ, так и наличие постоянной составляющей. Привлекательность кода NRZ, из-за которой имеет смысл заняться его улучшением, состоит в достаточно низкой частоте основной гармоники f0, которая равна N/2 Гц, как это было показано в предыдущем разделе. У других методов кодирования, например манчестерского, основная гармоника имеет более высокую частоту.

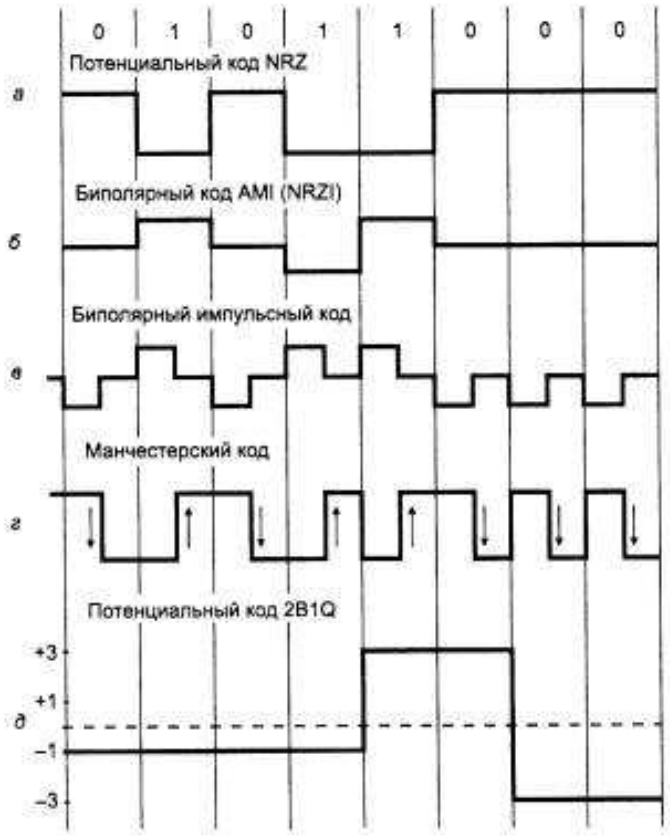


Рис. 2.17. Способы дискретного кодирования данных

Метод биполярного кодирования с альтернативной инверсией

Одной из модификаций метода NRZ является метод биполярного кодирования с альтернативной инверсией (Bipolar Alternate Mark Inversion, AMI). В этом методе (см. рис. 2.17, *б*) используются три уровня потенциала: отрицательный, нулевой и положительный. Для кодирования логического нуля используется нулевой потенциал, а логическая единица кодируется либо положительным потенциалом, либо отрицательным, при этом потенциал каждой новой единицы противоположен потенциалу предыдущей.

Код AMI частично ликвидирует проблемы постоянной составляющей и отсутствия самосинхронизации, присущие коду NRZ. Это происходит при передаче длинных последовательностей единиц. В этих случаях сигнал на линии представляет собой последовательность разнополярных импульсов с тем же спектром, что и у кода NRZ, передающего чередующиеся нули и единицы, т. е. без постоянной составляющей и с основной гармоникой N/2 Гц (где N – битовая скорость передачи данных). Длинные же последовательности нулей так же опасны для кода AMI, как и для кода NRZ – сигнал вырождается в постоянный потенциал нулевой амплитуды.

В коде AMI используются не два, а три уровня сигнала на линии. Дополнительный уровень требует увеличения мощности передатчика примерно на 3 дБ для обеспечения той же достоверности приема бит на линии, что является общим недостатком кодов с несколькими состояниями сигнала по сравнению с кодами, которые различают только два состояния.

Потенциальный код с инверсией при единице

Существует код, похожий на AMI, но только с двумя уровнями сигнала. При передаче нуля он 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 (Non Return to Zero with ones Inverted, NRZI). Этот код удобен в тех случаях, когда использование третьего уровня сигнала весьма нежелательно, например в оптических кабелях, где устойчиво распознаются два состояния сигнала: свет и темнота.

Биполярный импульсный код

Кроме потенциальных кодов в сетях используются и импульсные коды, когда данные представлены полным импульсом или же его частью – фронтом. Наиболее простым случаем такого подхода является биполярный импульсный код, в котором единица представлена импульсом одной полярности, а ноль – другой (см. рис. 2.17, *в*). Каждый импульс длится половину такта. Такой код обладает отличными самосинхронизирующими свойствами, но постоянная составляющая может присутствовать, например, при передаче длинной последовательности единиц или нулей. Кроме того, спектр у него шире, чем у потенциальных кодов. Так, при передаче всех нулей или единиц частота основной гармоники кода будет равна N Гц, что в два раза выше основной гармоники кода NRZ и в четыре раза выше основной гармоники кода AMI при передаче чередующихся единиц и нулей. Из-за слишком широкого спектра биполярный импульсный код используется редко.

Манчестерский код

В локальных сетях до недавнего времени самым распространенным методом кодирования был так называемый манчестерский код (см. рис. 2.17, *г*). Он применяется в технологиях Ethernet и Token Ring.

В манчестерском коде для кодирования единиц и нулей используется перепад потенциала, т. е. фронт импульса. При манчестерском кодировании каждый такт делится на две части. Информация кодируется перепадами потенциала, происходящими в середине каждого такта. Единица кодируется перепадом от низкого уровня сигнала к высокому, а ноль – обратным перепадом. В начале каждого такта может происходить служебный перепад сигнала, если нужно представить несколько единиц или нулей подряд. Так как сигнал изменяется, по крайней мере, один раз за такт передачи одного бита данных, то манчестерский код обладает хорошими самосинхронизирующими свойствами. Полоса пропускания манчестерского кода уже, чем у биполярного импульсного. У него также нет постоянной составляющей, а основная гармоника в худшем случае (при передаче последовательности единиц или нулей) имеет частоту N Гц, а в лучшем (при передаче чередующихся единиц и нулей) она равна N/2 Гц, как и у кодов AMI или NRZ. В среднем ширина полосы манчестерского кода в полтора раза уже, чем у биполярного импульсного кода, а основная гармоника колеблется вблизи значения 3N/4. Манчестерский код имеет еще одно преимущество перед биполярным импульсным кодом: в последнем для передачи данных используются три уровня сигнала, а в манчестерском – два.

Потенциальный код 2B1Q

На рис. 2.17, *д* показан потенциальный код с четырьмя уровнями сигнала для кодирования данных. Это код 2B1Q, название которого отражает его суть – каждые два бита (2B) передаются за один такт сигналом, имеющим четыре состояния (1Q), Паре бит 00 соответствует потенциал –2,5 В, паре бит 01 соответствует потенциал –0,833 В, паре 11 – потенциал +0,833 В, а паре 10 – потенциал +2,5 В. При этом способе кодирования требуются дополнительные меры по борьбе с длинными последовательностями одинаковых пар бит, т. к. при этом сигнал превращается в постоянную составляющую. 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. Таким образом, с помощью кода 2B1Q можно по одной и той же линии передавать данные в два раза быстрее, чем с помощью кода AMI или NRZI. Однако для его реализации мощность передатчика должна быть выше, чтобы четыре уровня четко различались приемником на фоне помех.

Логическое кодирование

Логическое кодирование используется для улучшения потенциальных кодов типа AMI, NRZI или 2Q1B. Логическое кодирование должно заменять длинные последовательности бит, приводящие к постоянному потенциалу, вкраплениями единиц. Как уже отмечалось выше, для логического кодирования характерны два метода: *избыточные коды* и *скрэмблирование*.

Избыточные коды основаны на разбиении исходной последовательности бит на порции, которые часто называют символами. Затем каждый исходный символ заменяется на новый, который имеет большее количество бит, чем исходный. Например, логический код 4B/5B, используемый в технологиях FDDI и Fast Ethernet, заменяет исходные символы длиной в 4 бита на символы длиной в 5 бит (табл. 2.1). Так как результирующие символы содержат избыточные биты, то общее количество битовых комбинаций в них больше, чем в исходных. Так, в коде 4B/5B результирующие символы могут содержать 32 битовых комбинации, в то время как исходные символы – только 16. Поэтому в результирующем коде можно отобрать 16 таких комбинаций, которые не содержат большого количества нулей, а остальные считать запрещенными кодами (*code violation*). Кроме устранения постоянной составляющей и придания коду свойства самосинхронизации, избыточные коды позволяют приемнику распознавать искаженные биты. Если приемник принимает запрещенный код, значит, на линии произошло искажение сигнала.

Таблица 2.1

Соответствие исходных и результирующих кодов 4B/5B

Исходный код	Результирующий код	Исходный код	Результирующий код
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

Код 4B/5B затем передается по линии с помощью физического кодирования одним из методов потенциального кодирования, чувствительным только к длинным последовательностям нулей. Символы кода 4B/5B длиной 5 бит гарантируют, что при любом их сочетании на линии не могут встретиться более трех нулей подряд.

Буква B в названии кода означает, что элементарный сигнал имеет два состояния – от английского *binary* – двоичный. Имеются также коды и с тремя состояниями сигнала, например, в коде 8B/6T для кодирования 8 бит исходной информации используется код из 6 сигналов, каждый из которых имеет три состояния. Избыточность кода 8B/6T выше, чем кода 4B/5B, т. к. на 256 исходных кодов приходится 3^6=729 результирующих символов.

Использование таблицы перекодировки является очень простой операцией, поэтому этот подход не усложняет сетевые адаптеры и интерфейсные блоки коммутаторов и маршрутизаторов.

Для обеспечения заданной пропускной способности линии передатчик, использующий избыточный код, должен работать с повышенной тактовой частотой. Так, для передачи кодов 4B/5B со скоростью 100 Мб/с передатчик должен работать с тактовой частотой 125 МГц.

Скрэмблирование. Перемешивание данных скрэмблером перед передачей их в линию с помощью потенциального кода является другим способом 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 следующее соотношение:

$$B_i = A_i \oplus B_{i-3} \oplus B_{i-5},$$

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 модулю 2).

Например, для исходной последовательности 110110000001 скрэмблер даст следующий результирующий код: $B_1 = A_1 = 1$ (первые три цифры результирующего кода будут совпадать с исходным, т. к. еще нет нужных предыдущих цифр).

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

Таким образом, на выходе скрэмблера появится последовательность 110001101111, в которой нет последовательности из шести нулей, присутствовавшей в исходном коде.

После получения результирующей последовательности приемник передает ее дескрэмблеру, который восстанавливает исходную последовательность на основании обратного соотношения:

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

Различные алгоритмы скрэмблирования отличаются количеством слагаемых, дающих цифру результирующего кода, и сдвигом между слагаемыми. Так, в сетях ISDN при передаче данных от сети к абоненту используется преобразование со сдвигами в 5 и 23 позиции, а при передаче данных от абонента в сеть – со сдвигами 18 и 23 позиции.

2.7. 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.

Наиболее существенными характеристиками метода передачи, а значит, и протокола, работающего на канальном уровне, являются следующие:

- асинхронный/синхронный;
- символьно-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 этих свойств характерны не только для протоколов канального уровня, но и для протоколов более высоких уровней.

Асинхронные протоколы

Асинхронные протоколы представляют собой наиболее старый способ связи. Эти протоколы оперируют не с кадрами, а с отдельными символами, которые представлены байтами со старт-стоповыми символами. Единицей передаваемых данных был не кадр данных, а отдельный символ. Некоторые символы имели управляющий характер, например символ <CR> предписывал телетайпу или дисплею выполнить возврат каретки на начало строки. В этих протоколах существуют управляющие последовательности, обычно начинающиеся с символа <ESC>. Эти последовательности вызывали на управляемом устройстве достаточно сложные действия, например, загрузку нового шрифта на принтер.

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

Постепенно асинхронные протоколы усложнялись и стали наряду с отдельными символами использовать целые блоки данных, то есть кадры. Например, популярный протокол XMODEM передает файлы между двумя компьютерами по асинхронному модему. Начало приема очередного блока файла инициируется символьной командой – принимающая сторона по-

стоянно передает символ ASCII NAK. Передающая сторона, приняв NAK, отправляет очередной блок файла, состоящий из 128 байт данных, заголовка и концевика. Заголовок состоит из специального символа SOH (Start Of Header) и номера блока. Концевик содержит контрольную сумму блока данных. Приемная сторона, получив новый блок, проверяла его номер и контрольную сумму. В случае совпадения этих параметров с ожидаемыми приемник отправлял символ АСК, а в противном случае – символ NAK, после чего передатчик должен был повторить передачу данного блока. В конце передачи файла передавался символ EOX.

Как видно из описания протокола XMODEM, часть управляющих операций выполнялась в асинхронных протоколах посылкой в асинхронном режиме отдельных символов, в то же время часть данных пересылалась блоками, что более характерно для синхронных протоколов.

Синхронные символьно-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 (рис. 2.18). Все биты кадра передаются непрерывным синхронным потоком, что значительно ускоряет передачу данных.

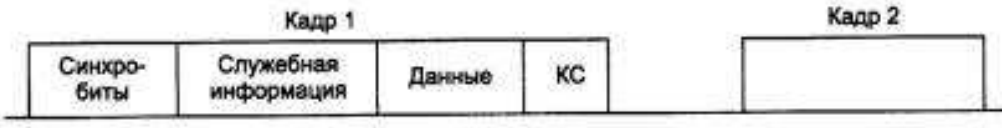


Рис. 2.18. Кадры синхронных протоколов

Так как байты в этих протоколах не отделяются друг от друга служебными сигналами, то одной из первых задач приемника является распознавание границ байтов. Затем приемник должен найти начало и конец кадра, а также определить границы каждого поля кадра: адреса назначения, адреса источника, других служебных полей заголовка, поля данных и контрольной суммы, если она имеется.

Большинство протоколов допускают использование в кадре поля данных переменной длины. Иногда и заголовок может иметь переменную длину. Обычно протоколы определяют максимальное значение, которое может иметь длина поля данных. Эта величина называется *максимальной единицей передачи данных (Maximum Transfer Unit, MTU)*. В некоторых

протоколах задается также минимальное значение, которое может иметь длина поля данных. Например, протокол Ethernet требует, чтобы поле данных содержало по крайней мере 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 также протоколы с кадрами фиксированной длины, например, в протоколе ATM кадры фиксированного размера 53 байт, включая служебную информацию. Для таких протоколов необходимо решить только первую часть задачи: распознать начало кадра.

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

Символьно-ориентированные протоколы

Символьно-ориентированные протоколы используются в основном для передачи блоков отображаемых символов, например текстовых файлов. Так как при синхронной передаче нет стоповых и стартовых битов, для синхронизации символов необходим другой метод. Синхронизация достигается за счет того, что передатчик добавляет два или более управляющих символа, называемых символами SYN, перед каждым блоком символов. В коде ASCII символ SYN имеет двоичное значение 0010110, 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, ASCII 0000010). Другой символ отмечает окончание кадра – ETX (End of TeXx, ASCII 0000011).

Однако такой простой способ выделения начала и конца кадра хорошо работал только в том случае, если внутри кадра не было символов STX и ETX. При подключении к компьютеру алфавитно-цифровых терминалов такая задача действительно не возникала. Тем не менее, синхронные символьно-ориентированные протоколы позднее стали использоваться и для

связи компьютера с компьютером, а в этом случае данные внутри кадра могут быть любые, если, например, между компьютерами передается программа. Наиболее популярным протоколом такого типа был протокол BSC компании IBM. Он работал в двух режимах: непрозрачном, в котором некоторые специальные символы внутри кадра запрещались, и прозрачном, в котором разрешалась передача внутри кадра любых символов, в том числе и ETX. Прозрачность достигалась за счет того, что перед управляющими символами STX и ETX всегда вставлялся символ DLE (Data Link Escape). Такая процедура называется стаффингом символов (*stuff* – всякая всячина, заполнитель). А если в поле данных кадра встречалась последовательность DLE ETX, то передатчик удваивал символ DLE, то есть порождал последовательность DLE DLE ETX. Приемник, встретив подряд два символа DLE DLE, всегда удалял первый, но оставшийся DLE уже не рассматривал как начало управляющей последовательности, т. е. оставшиеся символы DLE ETX считал просто пользовательскими данными.

Бит-ориентированные протоколы

Потребность в паре символов в начале и конце каждого кадра вместе с дополнительными символами DLE означает, что символьно-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. Так что этот метод допустим только с определенным типом кодировки, даже если кадр содержит чисто двоичные данные. Чтобы преодолеть эти проблемы, сегодня почти всегда используется более универсальный метод, называемый бит-ориентированной передачей. Этот метод сейчас применяется при передаче как двоичных, так и символьных данных.

На рис. 2.19 показаны три различные схемы бит-ориентированной передачи. Они отличаются способом обозначения начала и конца каждого кадра.

Первая схема, показанная на рис. 2.19, *а*, похожа на схему с символами STX и ETX в символьно-ориентированных протоколах. Начало и конец каждого кадра отмечается 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 8 битам.

Чтобы обеспечить синхронизацию приемника, передатчик посылает последовательность байтов простоя (каждый состоит из 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 бит-стаффинг. Схема вставки бита работает только во время передачи поля данных кадра. Если эта схема обнаруживает, что подряд передано пять 1, то она автоматически вставляет дополнительный 0 (даже если после этих пяти 1 шел 0). Поэтому последовательность 01111110 никогда не появится в поле данных кадра. Аналогичная схема работает в приемнике и выполняет обратную функцию. Когда после пяти 1 обнаруживается 0, он автоматически удаляется из поля данных кадра. Бит-стаффинг гораздо более экономичен, чем байт-стаффинг, так как вместо лишнего байта вставляется один бит.

Во второй схеме (см. рис. 2.19, б) для обозначения начала кадра имеется 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 этих сетях для обозначения факта незанятости среды в исходном состоянии по среде 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 играет 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. 2.19, *в*) использует для обозначения начала и конца кадра 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).

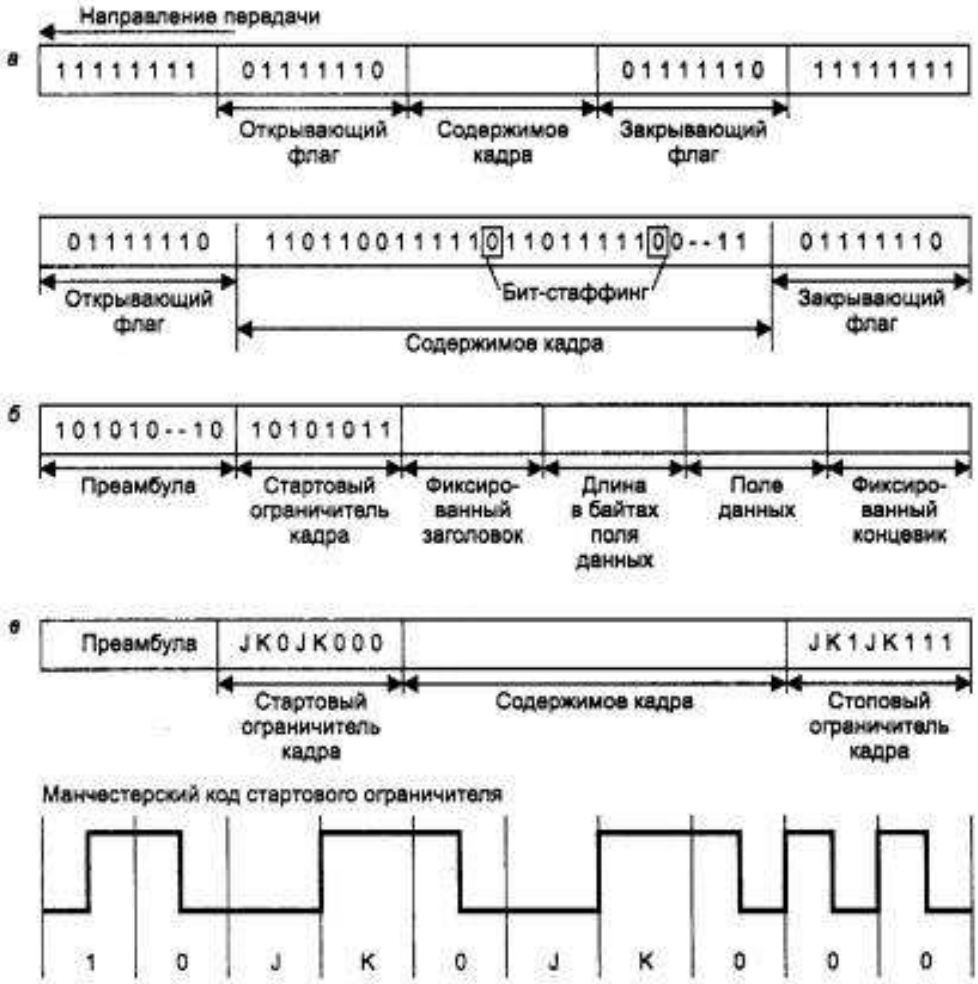


Рис. 2.19. Способы выделения начала и конца кадра при синхронной передаче

Начало кадра отмечается последовательностью JKOJKOOO, а конец – последовательностью JK1JK 100. Этот способ очень экономичен, т. к. не требует ни бит-стаффинга, ни поля длины, но его недостаток заключается в зависимости от принятого метода физического кодирования.

Передача с установлением соединения и без установления соединения

При передаче кадров данных на канальном уровне используются как дейтаграммные процедуры, работающие без становления соединения (connectionless), так и процедуры с предварительным установлением логического соединения (connection-oriented).

При дейтаграммной передаче кадр посылается в сеть «без предупреждения», и никакой ответственности за его утерю протокол не несет (рис. 2.20, *а*). Предполагается, что сеть всегда готова принять кадр от конечного узла. Дейтаграммный метод работает быстро, т. к. никаких предваритель-

ных 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.

Рис. 2.20. Протоколы без установления соединения (*а*) и с установлением соединения (*б*)

В этом случае узлу-получателю отправляется служебный кадр специального формата с предложением установить соединение (см. рис. 2.20, *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.

Заметим, что, 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) соединения, и информационные, переносящие собственно пользовательские данные.

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.

Бóльшая часть протоколов канального уровня выполняет только первую задачу – обнаружение ошибок, считая, что корректировать ошибки, т. е. повторно передавать данные, содержавшие искаженную информацию, должны протоколы верхних уровней. Так работают такие популярные протоколы локальных сетей, как 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, в которых не предусматриваются процедуры устранения ошибок, т.к. 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 случаются часто, то желательно уже на канальном уровне использовать протокол с коррекцией ошибок, а не оставлять эту работу протоколам верхних уровней. Протоколы верхних уровней, работая с большими тайм-аутами, восстановят потерянные данные с большой задержкой.

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 обнаружения ошибок

Все методы обнаружения ошибок основаны на передаче в составе кадра данных служебной избыточной информации, по которой можно судить с некоторой степенью вероятности о достоверности принятых дан-

ных. Эту служебную информацию принято называть *контрольной суммой* или *последовательностью контроля кадра (Frame Check Sequence, FCS)*. Контрольная сумма вычисляется как функция от основной информации, причем необязательно только путем суммирования. Принимающая сторона повторно вычисляет контрольную сумму кадра по известному алгоритму и в случае ее совпадения с контрольной суммой, вычисленной передающей стороной, делает вывод о том, что данные были переданы через сеть корректно.

Существует несколько распространенных алгоритмов вычисления контрольной суммы, отличающихся вычислительной сложностью и способностью обнаруживать ошибки в данных.

Контроль по паритету представляет собой наиболее простой метод контроля данных. Метод заключается в суммировании по модулю 2 всех бит контролируемой информации. Например, для данных 100101011 результатом контрольного суммирования будет значение 1. Результат суммирования также представляет собой один бит данных, который пересылается вместе с контролируемой информацией. При искажении при пересылке любого одного бита исходных данных (или контрольного разряда) результат суммирования будет отличаться от принятого контрольного разряда, что говорит об ошибке. Однако двойная ошибка, например 110101010, будет неверно принята за корректные данные. Поэтому контроль по паритету применяется к небольшим порциям данных, как правило, к каждому байту, что дает коэффициент избыточности для этого метода 1/8. Метод редко применяется в вычислительных сетях из-за его большой избыточности и невысоких диагностических способностей.

Циклический избыточный контроль (Cyclic Redundancy Check, CRC) является в настоящее время наиболее популярным методом контроля в вычислительных сетях. Метод основан на рассмотрении исходных данных в виде одного многоразрядного двоичного числа. Например, кадр стандарта Ethernet, состоящий из 1024 байт, будет рассматриваться как одно число, состоящее из 8192 бит. В качестве контрольной информации рассматривается остаток от деления этого числа на известный делитель R. Обычно в качестве делителя выбирается семнадцати- или тридцатитрехразрядное число, чтобы остаток от деления имел длину 16 разрядов (2 байта) или 32 разряда (4 байта). При получении кадра данных снова вычисляется остаток от деления на тот же делитель R, но при этом к данным кадра добавляется и содержащаяся в нем контрольная сумма. Если остаток от деления на R

равен нулю, то делается вывод об отсутствии ошибок в полученном кадре, в противном случае кадр считается искаженным.

Этот метод обладает более высокой вычислительной сложностью, но его диагностические возможности гораздо выше, чем у методов контроля по паритету. Метод обладает также невысокой степенью избыточности. Например, для кадра Ethernet размером в 1024 байт контрольная информация длиной в 4 байта составляет только 0,4 %.

Методы восстановления искаженных и потерянных кадров

Методы коррекции ошибок в вычислительных сетях основаны на повторной передаче кадра данных в том 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* – явное указание на то, что данный кадр нужно передать повторно.

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

Метод с простоями (Idle Source) 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. На рис. 2.21, *а* видно, что в этом случае производительность обмена данными существенно снижается – хотя передатчик и мог бы послать следующий кадр сразу же после отправки предыдущего, он обязан ждать прихода квитанции. Снижение производительности этого метода коррекции особенно заметно на низкоскоростных каналах связи, т. е. в территориальных сетях.

Второй метод называется *методом «скользящего окна» (sliding window)*. В этом методе 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. 2.21, *б* иллюстрирует этот метод для окна размером в W кадров. Здесь t_0 – исходный момент, t_1 и t_n – моменты прихода квитанций на первый и *n*-й кадр соответственно. Каждый раз, когда приходит квитанция, окно сдвигается влево, но его размер при этом не меняется и остается равным W. Заметим, что хотя в данном примере размер окна в процессе передачи остается постоянным, в реальных протоколах (например, TCP) можно встретить варианты данного алгоритма с изменяющимся размером окна.

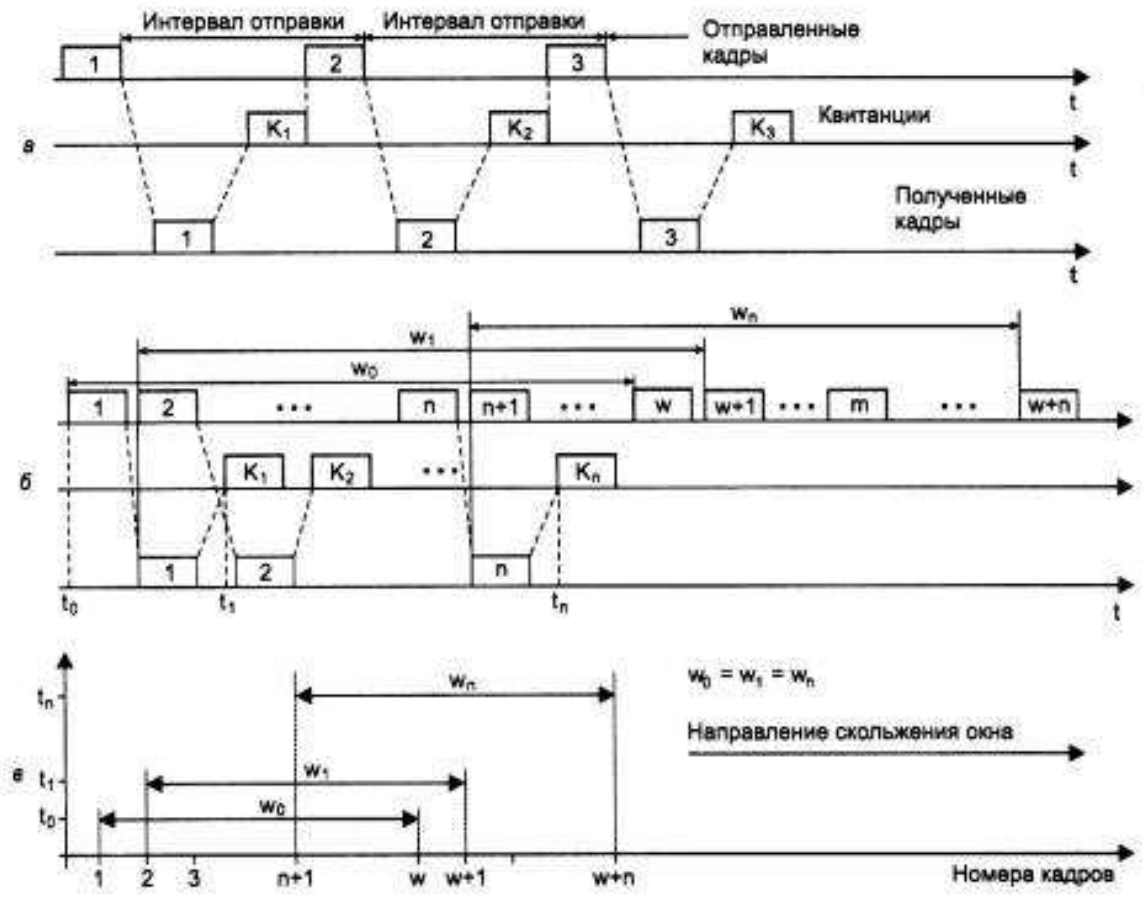


Рис. 2.21 Методы восстановления искаженных и потерянных кадров

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 он передается снова.

Если же поток квитанций поступает более-менее регулярно, в пределах допуска в W кадров, то скорость обмена достигает максимально возможной величины для данного канала и принятого протокола.

Метод скользящего окна более сложен в реализации, чем метод с простоями, т. к. 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.

Следует отметить, что в некоторых реализациях 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.

Метод скользящего окна реализован во многих протоколах: LLC2, LAP-B, X.25, TCP, Novell NCP Burst Mode.

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, когда кадры искажаются и теряются редко, для повышения скорости обмена данными размер окна нужно увеличивать, т. к. при этом передатчик будет посылать кадры с меньшими паузами. В ненадежных сетях размер окна следует уменьшать, т. к. при частых потерях и искажениях кадров резко возрастает объем вторично передаваемых через сеть кадров, а значит, пропускная способность сети будет расходоваться неэффективно. 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.

Компрессия данных

Компрессия (сжатие) данных применяется для сокращения времени их передачи. Так как на компрессию данных передающая сторона тратит дополнительное время, к которому нужно еще прибавить аналогичные затраты времени на декомпрессию этих данных принимающей стороной, то выгоды от сокращения времени на передачу сжатых данных обычно бывают заметны только для низкоскоростных каналов. Этот порог скорости

для современной аппаратуры составляет около 64 Кбит/с. Многие программные и аппаратные средства сети способны выполнять динамическую компрессию данных в отличие от статической, когда данные предварительно компрессируются, а уже затем отсылаются в сеть.

На практике может использоваться ряд алгоритмов компрессии, каждый из которых применим к определенному типу данных. Некоторые модемы (называемые интеллектуальными) предлагают *адаптивную компрессию*, при которой в зависимости от передаваемых данных выбирается определенный алгоритм компрессии. Рассмотрим некоторые из общих алгоритмов компрессии данных.

Десятичная упаковка. Когда данные состоят только из чисел, значительную экономию можно получить путем уменьшения количества используемых на цифру битов 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 используется, в частности, в рассмотренном выше методе цифрового кодирования голоса ADPCM, передающем в каждом такте только разницу между соседними замерами голоса.

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

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 встречаю-

щихся – кодами большей длины. Такое кодирование называется также статистическим кодированием. При статистическом кодировании коды выбираются таким образом, чтобы при анализе последовательности бит можно было бы однозначно определить соответствие определенной порции бит тому или иному символу или же запрещенной комбинации бит. Если данная последовательность бит представляет собой запрещенную комбинацию, то необходимо к ней добавить еще один бит и повторить анализ. Например, если при неравномерном кодировании для наиболее часто встречающегося символа «Р» выбран код 1, состоящий из одного бита, то значение 0 однобитного кода будет запрещенным. Иначе мы сможем закодировать только два символа. Для другого часто встречающегося символа «О» можно использовать код 01, а код 00 оставить как запрещенный. Тогда для символа «А» можно выбрать код 001, для символа «П» – код 0001 и т. п.

Вообще, неравномерное кодирование наиболее эффективно, когда неравномерность распределения частот передаваемых символов достаточно велика, как при передаче длинных текстовых строк. Напротив, при передаче двоичных данных, например кодов программ, оно малоэффективно, т. к. 8-битовые коды при этом распределены почти равномерно.

Многие модели коммуникационного оборудования, такие как модемы, мосты, коммутаторы и маршрутизаторы, поддерживают протоколы динамической компрессии, позволяющие сократить объем передаваемой информации в 4, а иногда и в 8 раз. В таких случаях говорят, что протокол обеспечивает коэффициент сжатия 1:4 или 1:8. Существуют стандартные протоколы компрессии, например V.42bis, а также большое количество нестандартных, фирменных протоколов. Реальный коэффициент компрессии зависит от типа передаваемых данных.

2.8. Методы коммутации: коммутация каналов и коммутация пакетов

Любые сети связи поддерживают некоторый способ коммутации своих абонентов между собой. Этими абонентами могут быть удаленные компьютеры, локальные сети, факс-аппараты или просто собеседники, общающиеся с помощью телефонных аппаратов. Практически невозможно предоставить каждой паре взаимодействующих абонентов свою собственную некоммутируемую физическую линию связи, которой они могли бы монопольно «владеть» в течение длительного времени. Поэтому в любой сети всегда применяется какой-либо способ коммутации абонентов, который обеспечивает доступность имеющихся физических каналов одновременно для нескольких сеансов связи между абонентами сети. На рис. 2.22 показана типичная структура сети с коммутацией абонентов.

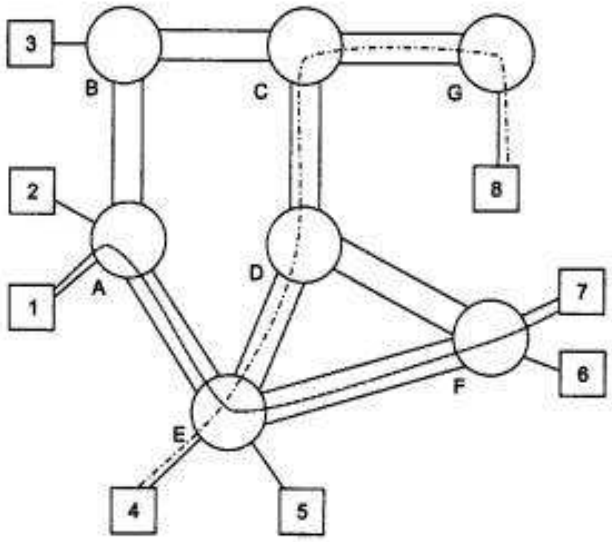


Рис. 2.22. Общая структура сети с коммутацией абонентов

Абоненты соединяются с коммутаторами индивидуальными линиями связи, каждая из которых используется в любой момент времени только одним, закрепленным за этой линией абонентом. Между коммутаторами линии связи разделяются несколькими абонентами, т. е. используются совместно.

Существуют две принципиально различные схемы коммутации абонентов в сетях: *коммутация каналов (circuit switching)* и *коммутация пакетов (packet switching)*. Внешне все эти схемы соответствуют приведенной на рис. 2.22 структуре сети, однако возможности и свойства их различны.

Каждая из этих схем имеет свои преимущества и недостатки, но по долгосрочным прогнозам многих специалистов будущее принадлежит технологии коммутации пакетов, как более гибкой и универсальной.

Как сети с коммутацией пакетов, так и сети с коммутацией каналов можно разделить на два класса по другому признаку: на сети с динамической коммутацией и сети с постоянной коммутацией.

В первом случае сеть разрешает устанавливать соединение по инициативе пользователя сети. Коммутация выполняется на время сеанса связи, а затем (опять же по инициативе одного из взаимодействующих пользователей) связь разрывается.

Во втором случае сеть не предоставляет пользователю возможность выполнить динамическую коммутацию с другим произвольным пользователем сети. Вместо этого сеть разрешает паре пользователей заказать соединение на длительный период времени. Соединение устанавливается не

пользователями, а персоналом, обслуживающим сеть. Время, на которое устанавливается постоянная коммутация, измеряется обычно несколькими месяцами. Режим постоянной коммутации в сетях с коммутацией каналов часто называется сервисом выделенных (*dedicated*) или арендуемых (*leased*) каналов.

Коммутация каналов

Коммутация каналов подразумевает образование непрерывного составного физического канала из последовательно соединенных отдельных канальных участков для прямой передачи данных между узлами. Отдельные каналы соединяются между собой специальной аппаратурой – коммутаторами, которые могут устанавливать связи между любыми конечными узлами сети. В сети с коммутацией каналов перед передачей данных всегда необходимо выполнить процедуру установления соединения, в процессе которой и создается составной канал.

Коммутаторы, а также соединяющие их каналы должны обеспечивать одновременную передачу данных нескольких абонентских каналов. Для этого они должны быть высокоскоростными и поддерживать какую-либо технику мультиплексирования абонентских каналов.

В настоящее время для мультиплексирования абонентских каналов используются две техники:

- техника частотного мультиплексирования (Frequency Division Multiplexing, FDM);
- техника мультиплексирования с разделением времени (Time Division Multiplexing, TDM).

Коммутация каналов на основе частотного мультиплексирования

Техника частотного мультиплексирования каналов (FDM) была разработана для телефонных сетей, но применяется она и для других видов сетей, например, сетей кабельного телевидения.

Рассмотрим особенности этого вида мультиплексирования на примере телефонной сети.

Речевые сигналы имеют спектр шириной примерно в 10000 Гц, однако основные гармоники укладываются в диапазон от 300 до 3400 Гц. Поэтому для качественной передачи речи достаточно образовать между двумя собеседниками канал с полосой пропускания в 3100 Гц, который и используется в телефонных сетях для соединения двух абонентов. В то же время полоса пропускания кабельных систем с промежуточными усилителями, соединяющих телефонные коммутаторы между собой, обычно со-

ставляет сотни килогерц, а иногда и сотни мегагерц. Однако непосредственно передавать сигналы нескольких абонентских каналов по широкополосному каналу невозможно, т. к. все они работают в одном и том же диапазоне частот и сигналы разных абонентов смешаются между собой так, что разделить их будет невозможно.

Для разделения абонентских каналов характерна техника модуляции высокочастотного несущего синусоидального сигнала низкочастотным речевым сигналом (рис. 2.23). Эта техника подобна технике аналоговой модуляции при передаче дискретных сигналов модемами, только вместо дискретного исходного сигнала используются непрерывные сигналы, порождаемые звуковыми колебаниями. В результате спектр модулированного сигнала переносится в другой диапазон, который располагается симметрично относительно несущей частоты и имеет ширину, приблизительно совпадающую с шириной модулирующего сигнала.

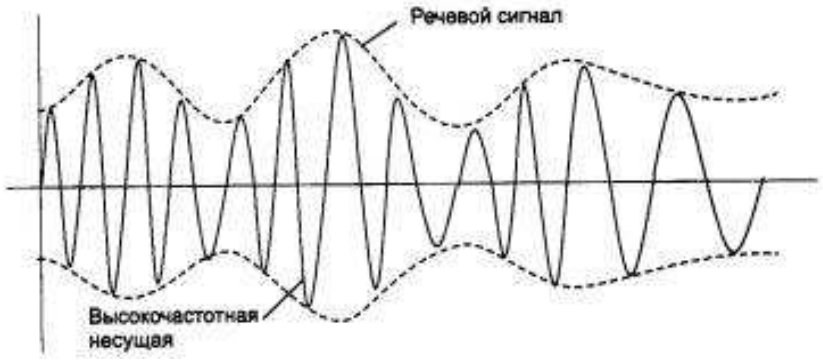


Рис. 2.23. Модуляция речевым сигналом

Если сигналы каждого абонентского канала перенести в свой собственный диапазон частот, то в одном широкополосном канале можно одновременно передавать сигналы нескольких абонентских каналов.

На входы FDM-коммутатора поступают исходные сигналы от абонентов телефонной сети. Коммутатор выполняет перенос частоты каждого канала в свой диапазон частот. Обычно высокочастотный диапазон делится на полосы, которые отводятся для передачи данных абонентских каналов (рис. 2.24). Чтобы низкочастотные составляющие сигналов разных каналов не смешивались между собой, полосы делают шириной в 4 кГц, а не в 3,1 кГц, оставляя между ними промежуток в 900 Гц. В канале между двумя FDM-коммутаторами одновременно передаются сигналы всех абонентских каналов, но каждый из них занимает свою полосу частот. Такой канал называют *уплотненным*.

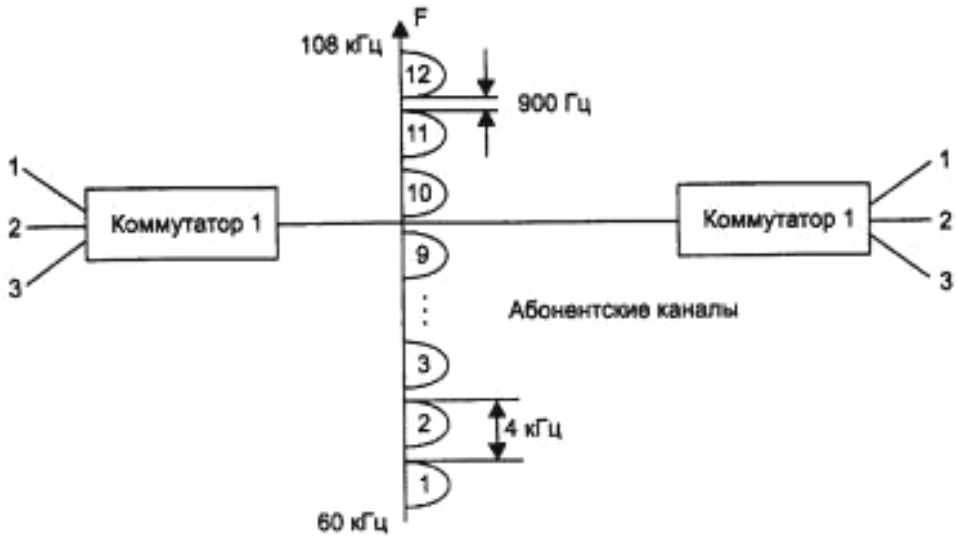


Рис. 2.24. Коммутация на основе частотного уплотнения

Выходной FDM-коммутатор выделяет модулированные сигналы каждой несущей частоты и передает их на соответствующий выходной канал, к которому непосредственно подключен абонентский телефон.

Коммутаторы FDM могут выполнять как динамическую, так и постоянную коммутацию. При динамической коммутации один абонент инициирует соединение с другим абонентом, посылая в сеть номер вызываемого абонента. Коммутатор динамически выделяет данному абоненту одну из свободных полос своего уплотненного канала. При постоянной коммутации за абонентом полоса в 4 кГц закрепляется на длительный срок путем настройки коммутатора по отдельному входу, недоступному пользователям.

Коммутация каналов на основе разделения времени

Коммутация на основе техники разделения частот разрабатывалась в расчете на передачу непрерывных сигналов, представляющих голос. При переходе к цифровой форме представления голоса была разработана новая техника мультиплексирования, ориентирующаяся на дискретный характер передаваемых данных. Эта техника носит название *мультиплексирования с разделением времени (Time Division Multiplexing, TDM)*. Реже используется и другое ее название – техника *синхронного режима передачи (Synchronous Transfer Mode, STM)*. Рисунок 2.25 поясняет принцип коммутации каналов на основе техники TDM.

Аппаратура TDM-сетей – мультиплексоры, коммутаторы, демультиплексоры – работает в режиме разделения времени, поочередно обслуживая в течение цикла своей работы все абонентские каналы. Цикл работы оборудования TDM равен 125 мкс, что соответствует периоду следования

замеров голоса в цифровом абонентском канале. Это значит, что мультиплексор или коммутатор успевает вовремя обслужить любой абонентский канал и передать его очередной замер далее по сети.

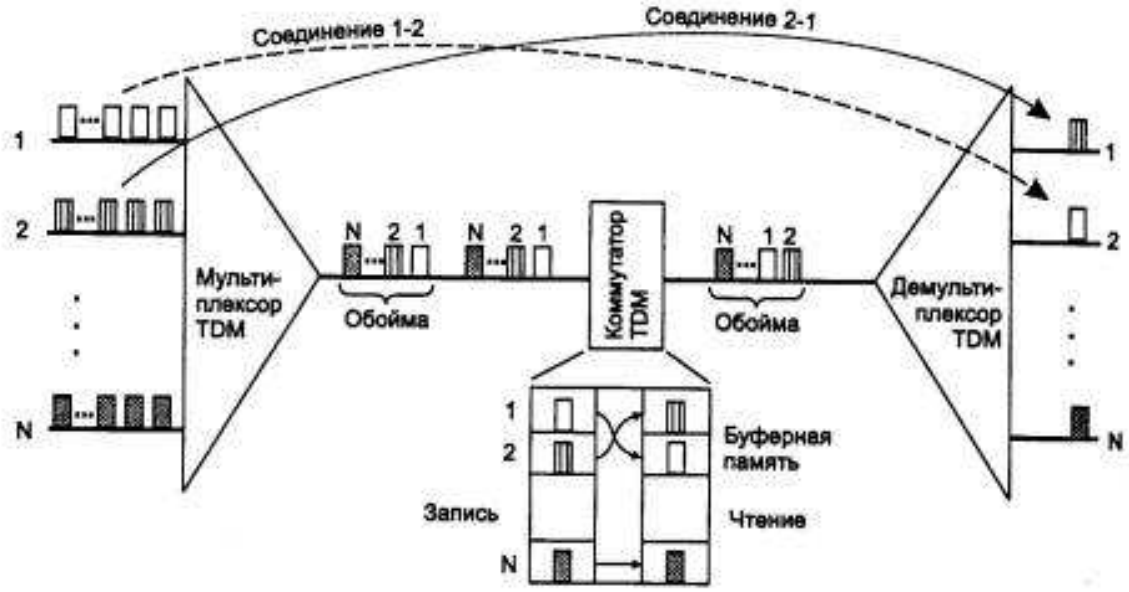


Рис. 2.25. Коммутация на основе разделения канала во времени

Каждому соединению выделяется один квант времени цикла работы аппаратуры, называемый также тайм-слотом. Длительность тайм-слота зависит от числа абонентских каналов, обслуживаемых мультиплексором TDM или коммутатором.

Мультиплексор принимает информацию по N входным каналам от конечных абонентов, каждый из которых передает данные по абонентскому каналу со скоростью 64 Кбит/с – 1 байт каждые 125 мкс. В каждом цикле мультиплексор выполняет следующие действия:

- прием от каждого канала очередного байта данных;
- составление из принятых байтов уплотненного кадра, называемого также обоймой;
- передача уплотненного кадра на выходной канал с битовой скоростью, равной $N \cdot 64$ Кбит/с.

Порядок байтов в обойме соответствует номеру входного канала, от которого этот байт получен. Количество обслуживаемых мультиплексором абонентских каналов зависит от его быстродействия.

Демультиплексор выполняет обратную задачу – он разбирает байты уплотненного кадра и распределяет их по своим нескольким выходным каналам, при этом он считает, что порядковый номер байта в обойме соответствует номеру выходного канала.

Коммутатор принимает уплотненный кадр по скоростному каналу от мультиплексора и записывает каждый байт из него в отдельную ячейку своей буферной памяти, причем в том порядке, в котором эти байты были упакованы в уплотненный кадр. Для выполнения операции коммутации байты извлекаются из буферной памяти не в порядке поступления, а в таком порядке, который соответствует поддерживаемым в сети соединениям абонентов. Так, например, если первый абонент левой части сети (см. рис. 2.25) должен соединиться со вторым абонентом в правой части сети, то байт, записанный в первую ячейку буферной памяти, будет извлекаться из нее вторым. «Перемешивая» нужным образом байты в обойме, коммутатор обеспечивает соединение конечных абонентов в сети.

Однажды выделенный номер тайм-слота остается в распоряжении соединения «входной канал – выходной слот» в течение всего времени существования этого соединения, даже если передаваемый трафик является пульсирующим и не всегда требует захваченного количества тайм-слотов. Это означает, что соединение в сети TDM всегда обладает известной и фиксированной пропускной способностью, кратной 64 Кбит/с.

Работа оборудования TDM напоминает работу сетей с коммутацией пакетов, т. к. каждый байт данных можно считать некоторым элементарным пакетом. Однако, в отличие от пакета компьютерной сети, «пакет» сети TDM не имеет индивидуального адреса. Его адресом является порядковый номер в обойме или номер выделенного тайм-слота в мультиплексоре или коммутаторе. Сети, использующие технику TDM, требуют синхронной работы всего оборудования, что и определило второе название этой техники – синхронный режим передач (STM). Нарушение синхронности разрушает требуемую коммутацию абонентов, т. к. при этом теряется адресная информация. Поэтому перераспределение тайм-слотов между различными каналами в оборудовании TDM невозможно, даже если в каком-то цикле работы мультиплексора тайм-слот одного из каналов оказывается избыточным, т. к. на входе этого канала в этот момент нет данных для передачи (например, абонент телефонной сети молчит).

Сети TDM могут поддерживать либо режим динамической коммутации, либо режим постоянной коммутации, а иногда и оба эти режима. Так, например, основным режимом цифровых телефонных сетей, работающих на основе технологии TDM, является динамическая коммутация, но они поддерживают также и постоянную коммутацию, предоставляя своим абонентам службу выделенных каналов.

Общие свойства сетей с коммутацией каналов

Сети с коммутацией каналов обладают несколькими важными общими свойствами независимо от того, какой тип мультиплексирования в них используется.

Сети с динамической коммутацией требуют предварительной процедуры установления соединения между абонентами. Для этого в сеть передается адрес вызываемого абонента, который проходит через коммутаторы и настраивает их на последующую передачу данных. Запрос на установление соединения маршрутизируется от одного коммутатора к другому и в конце концов достигает вызываемого абонента. Сеть может отказать в установлении соединения, если емкость требуемого выходного канала уже исчерпана. Для FDM-коммутатора емкость выходного канала равна количеству частотных полос этого канала, а для TDM-коммутатора – количеству тайм-слотов, на которые делится цикл работы канала. Сеть отказывает в соединении также в том случае, если запрашиваемый абонент уже установил соединение с кем-нибудь другим. В первом случае говорят, что занят коммутатор, а во втором – абонент. Возможность отказа в соединении является недостатком метода коммутации каналов.

Если соединение может быть установлено, то ему выделяется фиксированная полоса частот в FDM-сетях или же фиксированная пропускная способность в TDM-сетях. Эти величины остаются неизменными в течение всего периода соединения. Гарантированная пропускная способность сети после установления соединения является важным свойством, необходимым для таких приложений, как передача голоса, изображения или управления объектами в реальном масштабе времени. Однако динамически изменять пропускную способность канала по требованию абонента сети с коммутацией каналов не могут, что делает их неэффективными в условиях пульсирующего трафика.

Недостатком сетей с коммутацией каналов является невозможность применения пользовательской аппаратуры, работающей с разной скоростью. Отдельные части составного канала работают с одинаковой скоростью, т. к. сети с коммутацией каналов не буферизуют данные пользователей.

Сети с коммутацией каналов хорошо приспособлены для коммутации потоков данных постоянной скорости, когда единицей коммутации является не отдельный байт или пакет данных, а долговременный синхронный поток данных между двумя абонентами. Для таких потоков сети с коммутацией каналов добавляют минимум служебной информации для маршрутизации данных через сеть, используя временну́ю позицию каждого бита потока в качестве его адреса назначения в коммутаторах сети.

Обеспечение дуплексного режима работы на основе технологий FDM, TDM и WDM

Дуплексный режим – наиболее универсальный и производительный способ работы канала. Самым простым вариантом организации дуплексного режима является использование двух независимых физических каналов (двух пар проводников или двух световодов) в кабеле, каждый из которых работает в симплексном режиме, т. е. передает данные в одном направлении. Именно такая идея лежит в основе реализации дуплексного режима работы во многих сетевых технологиях, например Fast Ethernet или ATM.

Иногда такое простое решение оказывается недоступным или неэффективным. Чаще всего это происходит в тех случаях, когда для дуплексного обмена данными имеется всего один физический канал, а организация второго связана с большими затратами. Например, при обмене данными с помощью модемов через телефонную сеть у пользователя имеется только один физический канал связи с АТС – двухпроводная линия, и приобретать второй вряд ли целесообразно. В таких случаях дуплексный режим работы организуется на основе разделения канала на два логических подканала с помощью техники FDM или TDM.

Модемы для организации дуплексного режима работы на двухпроводной линии применяют технику FDM. Модемы, использующие частотную модуляцию, работают на четырех частотах: две частоты – для кодирования единиц и нулей в одном направлении, а остальные две частоты – для передачи данных в обратном направлении.

При цифровом кодировании дуплексный режим на двухпроводной линии организуется с помощью техники TDM. Часть тайм-слотов используется для передачи данных в одном направлении, а часть – для передачи в другом направлении. Обычно тайм-слоты противоположных направлений чередуются, из-за чего такой способ иногда называют «пинг-понговой» передачей. TDM-разделение линии характерно, например, для цифровых сетей с интеграцией услуг (ISDN) на абонентских двухпроводных окончаниях.

В волоконно-оптических кабелях при использовании одного оптического волокна для организации дуплексного режима работы применяется передача данных в одном направлении с помощью светового пучка одной длины волны, а в обратном – другой длины волны. Такая техника относится к методу FDM, однако для оптических кабелей она получила название *разделения по длине волны (Wave Division Multiplexing, WDM)*. WDM применяется и для повышения скорости передачи данных в одном направлении, обычно используя от 2 до 16 каналов.

Коммутация пакетов

Принципы коммутации пакетов

Коммутация пакетов – это техника коммутации абонентов, которая была специально разработана для эффективной передачи компьютерного трафика. Эксперименты по созданию первых компьютерных сетей на основе техники *коммутации каналов* показали, что этот вид коммутации не позволяет достичь высокой общей пропускной способности сети. Суть проблемы заключается в пульсирующем характере трафика, который генерируют типичные сетевые приложения. Например, при обращении к удаленному файловому серверу пользователь сначала просматривает содержимое каталога этого сервера, что порождает передачу небольшого объема данных. Затем он открывает требуемый файл в текстовом редакторе, и эта операция может создать достаточно интенсивный обмен данными, особенно если файл содержит объемные графические включения. После отображения нескольких страниц файла пользователь некоторое время работает с ними локально, что вообще не требует передачи данных по сети, а затем возвращает модифицированные копии страниц на сервер – и это снова порождает интенсивную передачу данных по сети.

Коэффициент пульсации трафика отдельного пользователя сети, равный отношению средней интенсивности обмена данными к максимально возможной, может составлять 1:50 или 1:100. Если для описанной сессии организовать коммутацию канала между компьютером пользователя и сервером, то бóльшую часть времени канал будет простаивать. В то же время коммутационные возможности сети будут использоваться – часть таймслотов или частотных полос коммутаторов будет занята и недоступна другим пользователям сети.

При коммутации пакетов все передаваемые пользователем сети сообщения разбиваются в исходном узле на сравнительно небольшие части, называемые пакетами. Напомним, что сообщением называется логически завершенная порция данных – запрос на передачу файла, ответ на этот запрос, содержащий весь файл, и т. п. Сообщения могут иметь произвольную длину, от нескольких байт до многих мегабайт. Пакеты тоже могут иметь переменную длину, но в узких пределах, например, от 46 до 1500 байт. Каждый пакет снабжается заголовком, в котором указывается адресная информация, необходимая для доставки пакета узлу назначения, а также номер пакета, который будет использоваться узлом назначения для сборки сообщения (рис. 2.26). Пакеты транспортируются в сети как независимые

информационные блоки. Коммутаторы сети принимают пакеты от конечных узлов и на основании адресной информации передают их друг другу, а в конечном итоге – узлу назначения.

Рис. 2.26. Разбиение сообщения на пакеты

Коммутаторы пакетной сети отличаются от коммутаторов каналов тем, что они имеют внутреннюю буферную память для временного хранения пакетов, если выходной порт коммутатора в момент принятия пакета занят передачей другого пакета (рис. 2.27). В этом случае пакет находится некоторое время в очереди пакетов в буферной памяти выходного порта, а когда до него дойдет очередь, то он передается следующему коммутатору. Такая схема передачи данных позволяет сглаживать пульсации трафика на магистральных связях между коммутаторами и тем самым использовать их наиболее эффективным образом для повышения пропускной способности сети в целом.

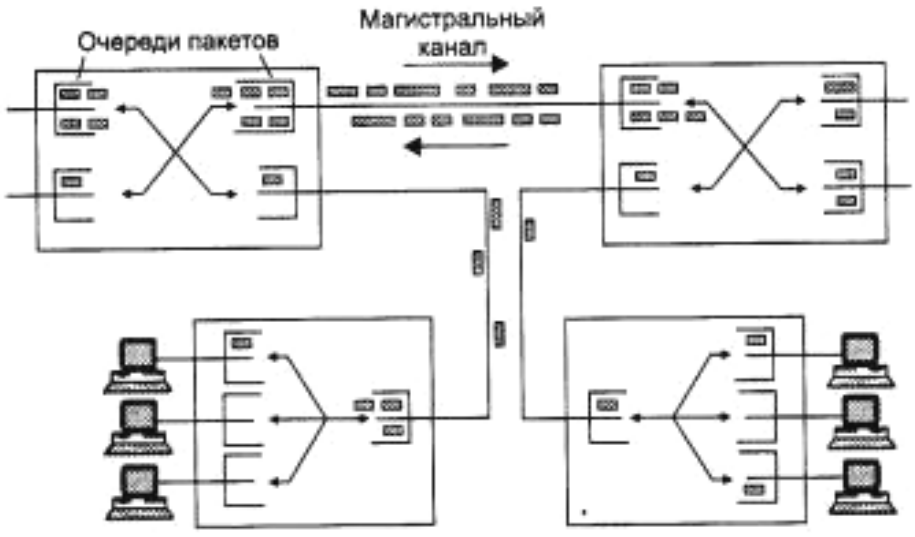


Рис. 2.27. Сглаживание пульсаций трафика в сети с коммутацией пакетов

Действительно, для пары абонентов наиболее эффективным было бы предоставление им в единоличное пользование скоммутированного канала связи, как это делается в сетях с коммутацией каналов. При этом способе время взаимодействия этой пары абонентов было бы минимальным, т. к. данные без задержек передавались бы от одного абонента другому. Простои канала во время пауз передачи абонентов не интересуют, для них важно быстрее решить свою собственную задачу. Сеть с коммутацией пакетов замедляет процесс взаимодействия конкретной пары абонентов, т. к. их пакеты могут ожидать в коммутаторах, пока по магистральным связям передаются другие пакеты, пришедшие в коммутатор ранее.

Тем не менее, общий объем передаваемых сетью компьютерных данных в единицу времени при технике коммутации пакетов будет выше, чем при технике коммутации каналов. Это происходит потому, что пульсации отдельных абонентов в соответствии с законом больших чисел распределяются во времени. Поэтому коммутаторы постоянно и достаточно равномерно загружены работой, если число обслуживаемых ими абонентов действительно велико. На рис. 2.27 показано, что трафик, поступающий от конечных узлов на коммутаторы, очень неравномерно распределен во времени. Однако коммутаторы более высокого уровня иерархии, которые обслуживают соединения между коммутаторами нижнего уровня, загружены более равномерно, и поток пакетов в магистральных каналах, соединяющих коммутаторы верхнего уровня, имеет почти максимальный коэффициент использования.

Более высокая эффективность сетей с коммутацией пакетов по сравнению с сетями с коммутацией каналов (при равной пропускной способности каналов связи) была доказана в 60-е годы как экспериментально, так и с помощью имитационного моделирования. Здесь уместна аналогия с мультипрограммными операционными системами. Каждая отдельная программа в такой системе выполняется дольше, чем в однопрограммной системе, когда программе выделяется все процессорное время, пока она не завершит свое выполнение. Однако общее число программ, выполняемых за единицу времени, в мультипрограммной системе больше, чем в однопрограммной.

Виртуальные каналы в сетях с коммутацией пакетов

Описанный выше режим передачи пакетов между двумя конечными узлами сети предполагает независимую маршрутизацию каждого пакета. Такой режим работы сети называется дейтаграммным, и при его использовании коммутатор может изменить маршрут какого-либо пакета в зависи-

мости от состояния сети – работоспособности каналов и других коммутаторов, длины очередей пакетов в соседних коммутаторах и т. п.

Существует и другой режим работы сети – передача пакетов по *виртуальному каналу* (*virtual circuit* или *virtual channel*). В этом случае перед тем, как начать передачу данных между двумя конечными узлами, должен быть установлен виртуальный канал, который представляет собой единственный маршрут, соединяющий эти конечные узлы. Виртуальный канал может быть динамическим или постоянным. Динамический виртуальный канал устанавливается при передаче в сеть специального пакета – запроса на установление соединения. Этот пакет проходит через коммутаторы и «прокладывает» виртуальный канал. Это означает, что коммутаторы запоминают маршрут для данного соединения и при поступлении последующих пакетов данного соединения отправляют их всегда по проложенному маршруту. Постоянные виртуальные каналы создаются администраторами сети путем ручной настройки коммутаторов.

При отказе коммутатора или канала на пути виртуального канала соединение разрывается и виртуальный канал нужно прокладывать заново. При этом он, естественно, обойдет отказавшие участки сети.

Каждый режим передачи пакетов имеет свои преимущества и недостатки. Дейтаграммный метод не требует предварительного установления соединения и поэтому работает без задержки перед передачей данных. Это особенно выгодно для передачи небольшого объема данных, когда время установления соединения может быть соизмеримым со временем передачи данных. Кроме того, дейтаграммный метод быстрее адаптируется к изменениям в сети.

При использовании метода виртуальных каналов время, затраченное на установление виртуального канала, компенсируется последующей быстрой передачей всего потока пакетов. Коммутаторы распознают принадлежность пакета к виртуальному каналу по специальной метке – номеру виртуального канала, а не анализируют адреса конечных узлов, как это делается при дейтаграммном методе.

Пропускная способность сетей с коммутацией пакетов

Одним из отличий метода коммутации пакетов от метода коммутации каналов является неопределенность пропускной способности соединения между двумя абонентами. В методе коммутации каналов после образования составного канала пропускная способность сети при передаче данных между конечными узлами известна: это пропускная способность канала. Данные после задержки, связанной с установлением канала, начинают

передаваться на максимальной для канала скорости (рис. 2.28, *а*). Время передачи сообщения в сети с коммутацией каналов $T_{к.к.}$ равно сумме задержки распространения сигнала по линии связи $t_{з.р.}$ и задержки передачи сообщения $t_{з.п.}$. Задержка распространения сигнала зависит от скорости распространения электромагнитных волн в конкретной физической среде, которая колеблется от 0,6 до 0,9 скорости света в вакууме. Время передачи сообщения равно *V*/*C*, где *V* – объем сообщения в битах, а *C* – пропускная способность канала в битах в секунду.

В сети с коммутацией пакетов наблюдается принципиально другая картина.

Процедура установления соединения в этих сетях, если она используется, занимает примерно такое же время, как и в сетях с коммутацией каналов, поэтому будем сравнивать только время передачи данных.

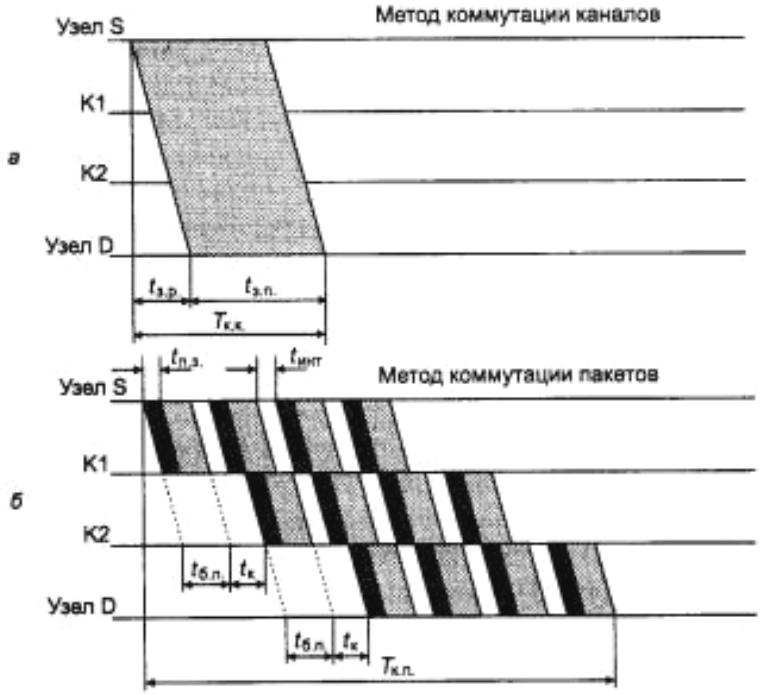


Рис. 2.28. Задержки передачи данных в сетях с коммутацией каналов (*а*) и пакетов (*б*)

На рис. 2.28, *б* показан пример передачи в сети с коммутацией пакетов. Предполагается, что в сеть передается сообщение того же объема, что и сообщение, иллюстрируемое рис. 2.31, *а*, однако оно разделено на пакеты, каждый из которых снабжен заголовком. Время передачи сообщения в сети с коммутацией пакетов обозначено на рисунке $T_{к.п.}$. При передаче этого сообщения, разбитого на пакеты, по сети с коммутацией пакетов возникают дополнительные временные задержки. Во-первых, это задержки в источнике передачи, который, помимо передачи собственно сообщения, тра-

тит дополнительное время на передачу заголовков $t_{п.з.}$, плюс к этому добавляются задержки $t_{инт}$, вызванные интервалами между передачей каждого следующего пакета (это время уходит на формирование очередного пакета стеком протоколов).

Во-вторых, дополнительное время тратится в каждом коммутаторе. Здесь задержки складываются из времени буферизации пакета $t_{б.п.}$ (коммутатор не может начать передачу пакета, не приняв его полностью в свой буфер) и времени коммутации $t_к$. Время буферизации равно времени приема пакета с битовой скоростью протокола. Время коммутации складывается из времени ожидания пакета в очереди и времени перемещения пакета в выходной порт. Если время перемещения пакета фиксировано и обычно невелико (от нескольких микросекунд до нескольких десятков микросекунд), то время ожидания пакета в очереди колеблется в очень широких пределах и заранее неизвестно, так как зависит от текущей загрузки сети пакетами.

Проведем грубую оценку задержки в передаче данных в сетях с коммутацией пакетов по сравнению с сетями с коммутацией каналов на простейшем примере. Пусть тестовое сообщение, которое нужно передать в обоих видах сетей, составляет 200 Кбайт. Отправитель находится от получателя на расстоянии 5000 км. Пропускная способность линий связи составляет 2 Мбит/с.

Время передачи данных по сети с коммутацией каналов складывается из времени распространения сигнала, которое для расстояния 5000 км можно оценить примерно в 25 мс, и времени передачи сообщения, которое при пропускной способности 2 Мбит/с и длине сообщения 200 Кбайт равно примерно 800 мс, то есть всего передача данных заняла 825 мс.

Оценим дополнительное время, которое потребуется для передачи этого сообщения по сети с коммутацией пакетов. Будем считать, что путь от отправителя до получателя пролегает через 10 коммутаторов. Исходное сообщение разбивается на пакеты в 1 Кбайт, всего 200 пакетов. Вначале оценим задержку, которая возникает в исходном узле. Предположим, что доля служебной информации, размещенной в заголовках пакетов, по отношению к общему объему сообщения составляет 10 %. Следовательно, дополнительная задержка, связанная с передачей заголовков пакетов, составляет 10 % от времени передачи целого сообщения, т. е. 80 мс. Если принять интервал между отправкой пакетов равным 1 мс, тогда дополнительные потери за счет интервалов составят 200 мс. Итого, в исходном уз-

ле из-за пакетирования сообщения при передаче возникла дополнительная задержка в 280 мс.

Каждый из 10 коммутаторов вносит задержку коммутации, которая может иметь большой разброс, от долей до тысяч миллисекунд. В данном примере примем, что на коммутацию в среднем тратится 20 мс. Кроме того, при прохождении сообщений через коммутатор возникает задержка буферизации пакета. Эта задержка при величине пакета 1 Кбайт и пропускной способности линии 2 Мбит/с равна 4 мс. Общая задержка, вносимая 10 коммутаторами, составит примерно 240 мс. В результате дополнительная задержка, созданная сетью с коммутацией пакетов, составила 520 мс. Учитывая, что вся передача данных в сети с коммутацией каналов заняла 825 мс, эту дополнительную задержку можно считать существенной.

Хотя приведенный расчет носит очень приблизительный характер, но он делает более понятными те причины, которые приводят к тому, что процесс передачи для определенной пары абонентов в сети с коммутацией пакетов является более медленным, чем в сети с коммутацией каналов.

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

На эффективность работы сети существенно влияют размеры пакетов, которые передает сеть. Слишком большие размеры пакетов приближают сеть с коммутацией пакетов к сети с коммутацией каналов, поэтому эффективность сети при этом падает. Слишком маленькие пакеты заметно увеличивают долю служебной информации, т. к. каждый пакет несет с собой заголовок фиксированной длины, а количество пакетов, на которые разбиваются сообщения, будет резко расти при уменьшении размера пакета. Существует некоторая золотая середина, которая обеспечивает максимальную эффективность работы сети, однако ее трудно определить точно, т. к. она зависит от многих факторов, некоторые из них к тому же постоянно меняются в процессе работы сети. Поэтому разработчики протоколов для сетей с коммутацией пакетов выбирают пределы, в которых может находиться длина пакета, а точнее, его поле данных, т. к. заголовок, как правило, имеет фиксированную длину. Обычно нижний предел поля данных выбирается равным нулю, что разрешает передавать служебные пакеты без пользовательских данных, а верхний предел не превышает 4-х килобайт. Приложения при передаче данных пытаются занять максимальный размер

поля данных, чтобы быстрее выполнить обмен данными, а небольшие пакеты обычно используются для квитанций о доставке пакета.

При выборе размера пакета необходимо учитывать также и интенсивность битовых ошибок канала. На ненадежных каналах необходимо уменьшать размеры пакетов, т. к. это уменьшает объем повторно передаваемых данных при искажениях пакетов.

2.9. Протоколы и стандарты локальных сетей. Структура стандартов IEEE 802.X

Общая характеристика протоколов локальных сетей

При организации взаимодействия узлов в локальных сетях основная роль отводится протоколу канального уровня. Однако для того, чтобы канальный уровень мог справиться с этой задачей, структура локальных сетей должна быть вполне определенной. Так, например, наиболее популярный протокол канального уровня – Ethernet – рассчитан на параллельное подключение всех узлов сети к общей для них шине – отрезку коаксиального кабеля или иерархической древовидной структуре сегментов, образованных повторителями. Протокол Token Ring также рассчитан на вполне определенную конфигурацию – соединение компьютеров в виде логического кольца.

Подобный подход, заключающийся в использовании простых структур кабельных соединений между компьютерами локальной сети, соответствовал основной цели, которую ставили перед собой разработчики первых локальных сетей во второй половине 70-х годов. Эта цель заключалась в нахождении простого и дешевого решения для объединения в вычислительную сеть нескольких десятков компьютеров, находящихся в пределах одного здания. Решение должно было быть недорогим, поскольку в сеть объединялись недорогие компьютеры – появившиеся и быстро распространившиеся тогда мини-компьютеры стоимостью в 10000 – 20000 долларов. Количество их в одной организации было небольшим, поэтому предел в несколько десятков (максимум – до сотни) компьютеров представлялся вполне достаточным для роста практически любой локальной сети.

Для упрощения и, соответственно, удешевления аппаратных и программных решений разработчики первых локальных сетей остановились на совместном использовании кабелей всеми компьютерами сети в режиме разделения времени, т. е. в режиме TDM. Наиболее явным образом режим совместного использования кабеля проявляется в классических сетях Ethernet, где коаксиальный кабель физически представляет собой недели-

мый отрезок кабеля, общий для всех узлов сети. Но и в сетях Token Ring и FDDI, где каждая соседняя пара компьютеров соединена, казалось бы, своими индивидуальными отрезками кабеля с концентратором, эти отрезки не могут использоваться компьютерами, которые непосредственно к ним подключены, в произвольный момент времени. Эти отрезки образуют логическое кольцо, доступ к которому как к единому целому может быть получен только по вполне определенному алгоритму, в котором участвуют все компьютеры сети. Использование кольца как общего разделяемого ресурса упрощает алгоритмы передачи по нему кадров, т. к. в каждый конкретный момент времени кольцо занято только одним компьютером.

Использование разделяемых сред (*shared media*) позволяет упростить логику работы сети. Например, отпадает необходимость контроля переполнения узлов сети кадрами от многих станций, решивших одновременно обменяться информацией. В глобальных сетях, где отрезки кабелей, соединяющих отдельные узлы, не рассматриваются как общий ресурс, такая необходимость возникает, и для решения этой проблемы в протоколы обмена информацией вводятся весьма сложные процедуры управления потоком кадров, предотвращающие переполнение каналов связи и узлов сети.

Использование в локальных сетях очень простых конфигураций (общая шина и кольцо) наряду с положительными имело и отрицательные последствия, из которых наиболее неприятными были ограничения по производительности и надежности. Наличие только одного пути передачи информации, разделяемого всеми узлами сети, в принципе ограничивало пропускную способность сети пропускной способностью этого пути (которая делилась в среднем на число компьютеров сети), а надежность сети – надежностью этого пути. Поэтому по мере повышения популярности локальных сетей и расширения их сфер применения все больше стали применяться специальные коммуникационные устройства – мосты и маршрутизаторы, – которые в значительной мере снимали ограничения единственной разделяемой среды передачи данных. Базовые конфигурации в форме общей шины и кольца превратились в элементарные структуры локальных сетей, которые можно теперь соединять друг с другом более сложным образом, образуя параллельные основные или резервные пути между узлами.

Тем не менее, внутри базовых структур по-прежнему работают все те же протоколы разделяемых единственных сред передачи данных, которые были разработаны более 15 лет назад. Это связано с тем, что хорошие скоростные и надежностные характеристики кабелей локальных сетей удовлетворяли в течение всех этих лет пользователей небольших компьютерных сетей, которые могли построить сеть без больших затрат только с помо-

щью сетевых адаптеров и кабеля. К тому же колоссальная инсталляционная база оборудования и программного обеспечения для технологий Ethernet и Token Ring способствовала тому, что сложился следующий подход: в пределах небольших сегментов используются старые протоколы в их неизменном виде, а объединение таких сегментов в общую сеть происходит с помощью дополнительного и достаточно сложного оборудования.

В последние несколько лет наметилось движение к отказу от разделяемых сред передачи данных в локальных сетях и переходу к применению активных коммутаторов, к которым конечные узлы присоединяются индивидуальными линиями связи. В чистом виде такой подход предлагается в технологии ATM (Asynchronous Transfer Mode), а в технологиях, носящих традиционные названия с приставкой *switched* (коммутируемый) (switched Ethernet, switched Token Ring, switched FDDI), обычно используется смешанный подход, сочетающий разделяемые и индивидуальные среды передачи данных. Чаще всего конечные узлы соединяются в небольшие разделяемые сегменты с помощью повторителей, а сегменты соединяются друг с другом с помощью индивидуальных коммутируемых связей.

Существует и достаточно заметная тенденция к использованию в традиционных технологиях так называемой микросегментации, когда даже конечные узлы сразу соединяются с коммутатором индивидуальными каналами. Такие сети дороже разделяемых или смешанных, но производительность их выше.

При использовании коммутаторов у традиционных технологий появился новый режим работы – *полнодуплексный (full-duplex)*. В разделяемом сегменте станции всегда работают в *полудуплексном режиме (half-duplex)*, т. к. в каждый момент времени сетевой адаптер станции либо передает свои данные, либо принимает чужие, но никогда не делает это одновременно. Это справедливо для всех технологий локальных сетей, т. к. разделяемые среды поддерживаются не только классическими технологиями локальных сетей Ethernet, Token Ring, FDDI, но и всеми новыми – Fast Ethernet, 100VG-AnyLAN, Gigabit Ethernet.

В полнодуплексном режиме сетевой адаптер может одновременно передавать свои данные в сеть и принимать из сети чужие данные. Такой режим несложно обеспечивается при прямом соединении с мостом/коммутатором или маршрутизатором, т. к. вход и выход каждого порта такого устройства работают независимо друг от друга, каждый со своим буфером кадров.

Сегодня каждая технология локальных сетей приспособлена для работы как в полудуплексном, так и полнодуплексном режимах. В этих режимах ограничения, накладываемые на общую длину сети, существенно

отличаются, так что одна и та же технология может позволять строить весьма различные сети в зависимости от выбранного режима работы (который зависит от того, какие устройства используются для соединения узлов – повторители или коммутаторы). Например, технология Fast Ethernet позволяет для полудуплексного режима строить сети диаметром не более 200 метров, а для полнодуплексного режима ограничений на диаметр сети не существует. Поэтому при сравнении различных технологий необходимо обязательно принимать во внимание возможность их работы в двух режимах.

Структура стандартов IEEE 802.X

В 1980 году в институте IEEE был организован комитет 802 по стандартизации локальных сетей, в результате работы которого было принято семейство стандартов IEEE 802-x, которые содержат рекомендации по проектированию нижних уровней локальных сетей. Позже результаты работы этого комитета легли в основу комплекса международных стандартов ISO 8802-1...5. Эти стандарты были созданы на основе очень распространенных фирменных стандартов сетей Ethernet, ArcNet и Token Ring.

Помимо IEEE в работе по стандартизации протоколов локальных сетей принимали участие и другие организации. Так, для сетей, работающих на оптоволокне, американским институтом по стандартизации ANSI был разработан стандарт FDDI, обеспечивающий скорость передачи данных 100 Мб/с. Работы по стандартизации протоколов ведутся также ассоциацией ECMA, которой приняты стандарты ECMA-80, 81, 82 для локальной сети типа Ethernet и впоследствии – стандарты ECMA-89,90 по методу передачи маркера.

Стандарты семейства IEEE 802.X охватывают только два нижних уровня семиуровневой модели OSI – физический и канальный. Это связано с тем, что именно эти уровни в наибольшей степени отражают специфику локальных сетей. Старшие же уровни, начиная с сетевого, в значительной степени имеют общие черты как для локальных, так и для глобальных сетей.

Специфика локальных сетей также нашла свое отражение в разделении канального уровня на два подуровня, которые часто называют также уровнями. Канальный уровень (Data Link Layer) делится в локальных сетях на два подуровня:

- логической передачи данных (Logical Link Control, LLC);
- управления доступом к среде (Media Access Control, MAC).

Уровень MAC появился из-за существования в локальных сетях разделяемой среды передачи данных. Именно этот уровень обеспечивает корректное совместное использование общей среды, предоставляя ее в соответствии с определенным алгоритмом в распоряжение той или иной станции сети. После того как доступ к среде получен, ею может пользоваться более высокий уровень – уровень LLC, организующий передачу логических единиц данных, кадров информации с различным уровнем качества транспортных услуг. В современных локальных сетях получили распространение несколько протоколов уровня MAC, реализующих различные алгоритмы доступа к разделяемой среде. Эти протоколы полностью определяют специфику таких технологий, как Ethernet, Fast Ethernet, Gigabit Ethernet, Token Ring, FDDI, l00VG-AnyLAN.

Уровень LLC отвечает за передачу кадров данных между узлами с различной степенью надежности, а также реализует функции интерфейса с прилегающим к нему сетевым уровнем. Именно через уровень LLC сетевой протокол запрашивает у канального уровня нужную ему транспортную операцию с нужным качеством. На уровне LLC существуют несколько режимов работы, отличающихся наличием или отсутствием на этом уровне процедур восстановления кадров в случае их потери или искажения, т. е. отличающихся качеством транспортных услуг этого уровня.

Протоколы уровней MAC и LLC взаимно независимы: каждый протокол уровня MAC может применяться с любым протоколом уровня LLC, и наоборот.

Стандарты IEEE 802 имеют достаточно четкую структуру, приведенную на рис. 2.29.

Эта структура появилась в результате большой работы, проведенной комитетом 802 по выделению в разных фирменных технологиях общих подходов и общих функций, а также согласованию стилей их описания. В результате канальный уровень был разделен на два упомянутых подуровня. Описание каждой технологии разделено на две части: описание уровня MAC и описание физического уровня. Как видно из рисунка, практически у каждой технологии единственному протоколу уровня MAC соответствуют несколько вариантов протоколов физического уровня (на рисунке в целях экономии места приведены только технологии Ethernet и Token Ring, но все сказанное справедливо также и для остальных технологий, таких как ArcNet, FDDI, l00VG-AnyLAN).

Над канальным уровнем всех технологий изображен общий для них протокол LLC, поддерживающий несколько режимов работы, но независимый от выбора конкретной технологии. Стандарт LLC курирует подкомитет 802.2. Даже технологии, стандартизованные не в рамках комитета 802,

ориентируются на использование протокола LLC, определенного стандартом 802.2, например протокол FDDI, стандартизованный ANSI.

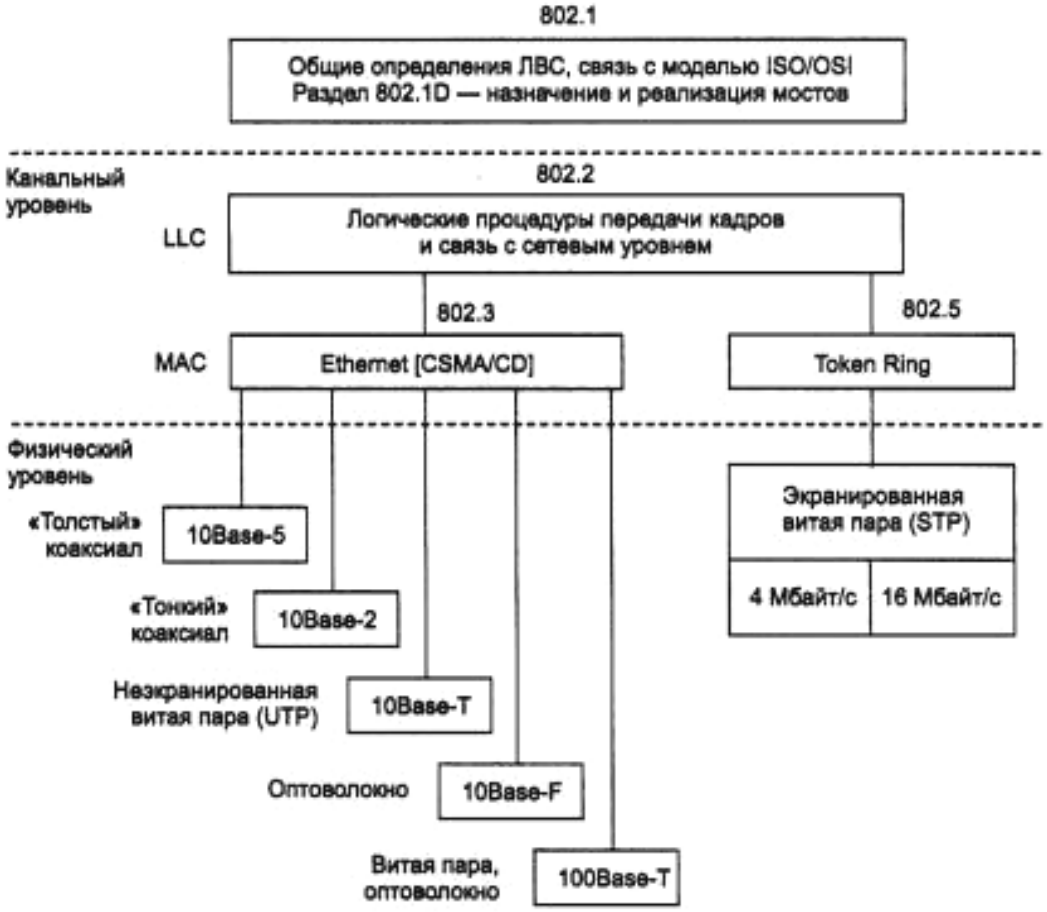


Рис. 2.29. Структура стандартов IEEE 802.X

Особняком стоят стандарты, разрабатываемые подкомитетом 802.1. Эти стандарты носят общий для всех технологий характер. В подкомитете 802.1 были разработаны общие определения локальных сетей и их свойств, определена связь трех уровней модели IEEE 802 с моделью OSI. Но наиболее практически важными являются стандарты 802.1, которые описывают взаимодействие между собой различных технологий, а также стандарты по построению более сложных сетей на основе базовых топологий. Эта группа стандартов носит общее название стандартов межсетевого взаимодействия (*internetworking*). Сюда входят такие важные стандарты, как стандарт 802. ID, описывающий логику работы моста/коммутатора, стандарт 802.1H, определяющий работу транслирующего моста, который может без маршрутизатора объединять сети Ethernet и FDDI, Ethernet и Token Ring, и т. п. Сегодня набор стандартов, разработанных подкомитетом 802.1, продолжает расти. Например, недавно он пополнился важным стандартом

802.1Q, определяющим способ построения виртуальных локальных сетей VLAN в сетях на основе коммутаторов.

Стандарты 802.3,802.4,802.5 и 802.12 описывают технологии локальных сетей, которые появились в результате улучшений фирменных технологий, легших в их основу. Так, основу стандарта 802.3 составила технология Ethernet, разработанная компаниями Digital, Intel и Xerox (или Ethernet DIX), стандарт 802.4 появился как обобщение технологии ArcNet компании Datapoint Corporation, а стандарт 802.5 в основном соответствует технологии Token Ring компании IBM.

Исходные фирменные технологии и их модифицированные варианты – стандарты 802.x – в ряде случаев долгие годы существовали параллельно. Например, технология ArcNet так до конца не была приведена в соответствие со стандартом 802.4 (теперь это делать поздно, т. к. примерно с 1993 года производство оборудования ArcNet было свернуто). Расхождения между технологией Token Ring и стандартом 802.5 тоже периодически возникают, т. к. компания IBM регулярно вносит усовершенствования в свою технологию и комитет 802.5 отражает эти усовершенствования в стандарте с некоторым запозданием. Исключение составляет технология Ethernet. Последний фирменный стандарт Ethernet DIX был принят в 1980 году, и с тех пор никто больше не предпринимал попыток фирменного развития Ethernet. Все новшества в семействе технологий Ethernet вносятся только в результате принятия открытых стандартов комитетом 802.3.

Более поздние стандарты изначально разрабатывались не одной компанией, а группой заинтересованных компаний, а потом передавались в соответствующий подкомитет IEEE 802 для утверждения. Так произошло с технологиями Fast Ethernet, l00VG-AnyLAN, Gigabit Ethernet. Группа заинтересованных компаний образовывала сначала небольшое объединение, а затем по мере развития работ к нему присоединялись другие компании, так что процесс принятия стандарта носил открытый характер.

Сегодня комитет 802 включает следующий ряд подкомитетов, в который входят как уже упомянутые, так и некоторые другие:

- 802.1 – Internetworking – объединение сетей;
- 802.2 – Logical Link Control, LLC – управление логической передачей данных;
- 802.3 – Ethernet с методом доступа CSMA/CD;
- 802.4 – Token Bus LAN – локальные сети с методом доступа Token Bus;
- 802.5 – Token Ring LAN – локальные сети с методом доступа Token Ring;
- 802.6 – Metropolitan Area Network, MAN – сети мегаполисов;

– 802.7 – Broadband Technical Advisory Group – техническая консультационная группа по широкополосной передаче;
– 802,8 – Fiber Optic Technical Advisory Group – техническая консультационная группа по волоконно-оптическим сетям;
– 802.9 – Integrated Voice and data Networks – интегрированные сети передачи голоса и данных;
– 802.10 – Network Security – сетевая безопасность;
– 802.11 – Wireless Networks – беспроводные сети;
– 802.12 – Demand Priority Access LAN, l00VG-AnyLAN – локальные сети с методом доступа по требованию с приоритетами.

2.10. Протокол LLC уровня управления логическим каналом

Протокол LLC обеспечивает для технологий локальных сетей нужное качество услуг транспортной службы, передавая свои кадры либо дейтаграммным способом, либо с помощью процедур с установлением соединения и восстановлением кадров. Протокол LLC занимает уровень между сетевыми протоколами и протоколами уровня MAC. Протоколы сетевого уровня передают через межуровневый интерфейс данные для протокола LLC – свой пакет (например, пакет IP, IPX или NetBEUI), адресную информацию об узле назначения, а также требования к качеству транспортных услуг, которое протокол LLC должен обеспечить. Протокол LLC помещает пакет протокола верхнего уровня в свой кадр, который дополняется необходимыми служебными полями. Далее через межуровневый интерфейс протокол LLC передает свой кадр вместе с адресной информацией об узле назначения соответствующему протоколу уровня MAC, который упаковывает кадр LLC в свой кадр (например, кадр Ethernet).

В основу протокола LLC положен протокол HDLC (High-level Data Link Control Procedure), являющийся стандартом ISO. Собственно стандарт HDLC представляет собой обобщение нескольких близких стандартов, характерных для различных технологий: протокола LAP-B сетей X.25 (стандарта, широко распространенного в территориальных сетях), LAP-D, используемого в сетях ISDN, LAP-M, работающего в современных модемах. В спецификации IEEE 802.2 также имеется несколько небольших отличий от стандарта HDLC.

Первоначально в фирменных технологиях подуровень LLC не выделялся в самостоятельный подуровень, да и его функции растворялись в общих функциях протокола канального уровня. Из-за больших различий в функциях протоколов фирменных технологий, которые можно отнести к

уровню LLC, на уровне LLC пришлось ввести три типа процедур. Протокол сетевого уровня может обращаться к одной из этих процедур.

Три типа процедур уровня LLC

В соответствии со стандартом 802.2 уровень управления логическим каналом LLC предоставляет верхним уровням три типа процедур:

– LLC1 – процедура без установления соединения и без подтверждения;

– LLC2 – процедура с установлением соединения и подтверждением;

– LLC3 – процедура без установления соединения, но с подтверждением.

Этот набор процедур является общим для всех методов доступа к среде, определенных стандартами 802.3 - 802.5, а также стандартом FDDI и стандартом 802.12 на технологию l00VG-AnyLAN.

Процедура без установления соединения и без подтверждения LLC1 дает пользователю средства для передачи данных с минимумом издержек. Это дейтаграммный режим работы. Обычно этот вид процедуры используется, когда такие функции, как восстановление данных после ошибок и упорядочивание данных, выполняются протоколами вышележащих уровней, поэтому нет нужды дублировать их на уровне LLC.

Процедура с установлением соединений и подтверждением LLC2 дает пользователю возможность установить логическое соединение перед началом передачи любого блока данных и, если это требуется, выполнить процедуры восстановления после ошибок и упорядочивание потока этих блоков в рамках установленного соединения. Протокол LLC2 во многом аналогичен протоколам семейства HDLC (LAP-B, LAP-D, LAP-M), которые применяются в глобальных сетях для обеспечения надежной передачи кадров на зашумленных линиях. Протокол LLC2 работает в режиме скользящего окна.

В некоторых случаях (например, при использовании сетей в системах реального времени, управляющих промышленными объектами), когда временные издержки установления логического соединения перед отправкой данных неприемлемы, а подтверждение о корректности приема переданных данных необходимо, базовая процедура без установления соединения и без подтверждения не подходит. Для таких случаев предусмотрена дополнительная процедура, называемая *процедурой без установления соединения, но с подтверждением LLC3*.

Использование одного из трех режимов работы уровня LLC зависит от стратегии разработчиков конкретного стека протоколов. Например, в стеке TCP/IP уровень LLC всегда работает в режиме LLC1, выполняя простую работу извлечения из кадра и демультиплексирования пакетов различных протоколов – IP, ARP, RARP. Аналогично используется уровень LLC стеком IPX/SPX.

Структура кадров LLC. Процедура с восстановлением кадров LLC2

По своему назначению все кадры уровня LLC (называемые в стандарте 802.2 блоками данных – Protocol Data Unit, PDU) подразделяются на три типа: информационные, управляющие и ненумерованные.

Информационные кадры (Information) предназначены для передачи информации в процедурах с установлением логического соединения LLC2 и должны обязательно содержать поле информации. В процессе передачи информационных блоков осуществляется их нумерация в режиме скользящего окна.

Управляющие кадры (Supervisory) предназначены для передачи команд и ответов в процедурах с установлением логического соединения LLC2, в том числе запросов на повторную передачу искаженных информационных блоков.

Ненумерованные кадры (Unnumbered) предназначены для передачи ненумерованных команд и ответов, выполняющих в процедурах без установления логического соединения передачу информации, идентификацию и тестирование LLC-уровня, а в процедурах с установлением логического соединения LLC2 – установление и разъединение логического соединения, а также информирование об ошибках.

Все типы кадров уровня LLC имеют единый формат, приведенный на рис. 2.30.

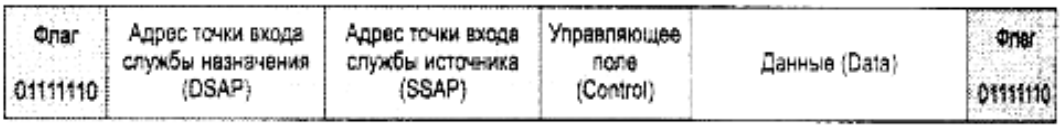


Рис. 2.30. Формат кадров LLC

Кадр LLC обрамляется двумя однобайтовыми полями «Флаг», имеющими значение 01111110. Флаги используются на уровне MAC для определения границ кадра LLC. В соответствии с многоуровневой структурой протоколов стандартов IEEE 802, кадр LLC вкладывается в кадр уровня

MAC – кадр Ethernet, Token Ring, FDDI и т. д. При этом флаги кадра LLC отбрасываются.

Кадр LLC содержит поле данных и заголовок, который состоит из трех полей:

– адрес точки входа службы назначения (Destination Service Access Point, DSAP);

– адрес точки входа службы источника (Source Service Access Point, SSAP);

– управляющее поле (Control).

Поле данных кадра LLC предназначено для передачи по сети пакетов протоколов вышележащих уровней – сетевых протоколов IP, IPX, AppleTalk, DECnet, в редких случаях – прикладных протоколов, когда те вкладывают свои сообщения непосредственно в кадры канального уровня. Поле данных может отсутствовать в управляющих кадрах и некоторых ненумерованных кадрах.

Адресные поля DSAP и SSAP занимают по 1 байту. Они позволяют указать, какая служба верхнего уровня пересылает данные с помощью этого кадра. Программному обеспечению узлов сети при получении кадров канального уровня необходимо распознать, какой протокол вложил свой пакет в поле данных поступившего кадра, чтобы передать извлеченный из кадра пакет нужному протоколу верхнего уровня для последующей обработки. Для идентификации этих протоколов вводятся так называемые адреса точки входа службы (Service Access Point, SAP). Значения адресов SAP приписываются протоколам в соответствии со стандартом 802.2. Например, для протокола IP значение SAP равно 0x6, для протокола NetBIOS – 0xF0. Для одних служб определена только одна точка входа и, соответственно, только один SAP, а для других – несколько, когда адреса DSAP и SSAP совпадают. Например, если в кадре LLC значения DSAP и SSAP содержат код протокола IPX, то обмен кадрами осуществляется между двумя IPX-модулями, выполняющимися в разных узлах. Но в некоторых случаях в кадре LLC указываются различающиеся DSAP и SSAP. Это возможно только в тех случаях, когда служба имеет несколько адресов SAP, что может быть использовано протоколом узла отправителя в специальных целях, например, для уведомления узла получателя о переходе протокола-отправителя в некоторый специфический режим работы. Этим свойством протокола LLC часто пользуется протокол NetBEUI.

Поле управления (1 или 2 байта) имеет сложную структуру при работе в режиме LLC2 и достаточно простую структуру при работе в режиме LLC1 (рис. 2.31).

Рис. 2.31. Структура поля управления

В режиме LLC1 используется только один тип кадра – ненумерованный. У этого кадра поле управления имеет длину в один байт. Все подполя поля управления ненумерованных кадров принимают нулевые значения, так что значимыми остаются только первые два бита поля, используемые как признак типа кадра. Учитывая, что в протоколе Ethernet при записи реализован обратный порядок бит в байте, то запись поля управления кадра LLC1, вложенного в кадр протокола Ethernet, имеет значение 0x03 (здесь и далее префикс 0x обозначает шестнадцатеричное представление).

В режиме LLC2 используются все три типа кадров. В этом режиме кадры делятся на команды и ответы на эти команды. Бит P/F (Poll/Final) имеет следующее значение: в командах он называется битом Poll и требует, чтобы на команду был дан ответ, а в ответах он называется битом Final и говорит о том, что ответ состоит из одного кадра.

Ненумерованные кадры используются на начальной стадии взаимодействия двух узлов, т.е. на стадии установления соединения по протоколу LLC2. Поле M ненумерованных кадров определяет несколько типов команд, которыми пользуются два узла на этапе установления соединения. Ниже приведены примеры некоторых команд.

Установить сбалансированный асинхронный расширенный режим (SABME). Эта команда является запросом на установление соединения. Она является одной из команд полного набора команд такого рода протокола HDLC. Расширенный режим означает использование двухбайтных полей управления для кадров остальных двух типов.

Ненумерованное подтверждение (UA). Служит для подтверждения установления или разрыва соединения.

Сброс соединения (REST). Запрос на разрыв соединения.

После установления соединения данные и положительные квитанции начинают передаваться в информационных кадрах. Логический канал протокола LLC2 является дуплексным, так что данные могут передаваться в обоих направлениях. Если поток дуплексный, то положительные квитан-

ции на кадры также доставляются в информационных кадрах. Если же потока кадров в обратном направлении нет или же нужно передать отрицательную квитанцию, то используются супервизорные кадры.

В информационных кадрах имеется поле N(S) для указания номера отправленного кадра, а также поле N(R) для указания номера кадра, который приемник ожидает получить от передатчика следующим. При работе протокола LLC2 используется скользящее окно размером в 127 кадров, а для их нумерации циклически используется 128 чисел, от 0 до 127.

Приемник всегда помнит номер последнего кадра, принятого от передатчика, и поддерживает переменную с указанным номером кадра, который он ожидает принять от передатчика следующим. Обозначим его через V(R). Именно это значение передается в поле N(R) кадра, посылаемого передатчику. Если в ответ на этот кадр приемник принимает кадр, в котором номер посланного кадра N(S) совпадает с номером ожидаемого кадра V(R), то такой кадр считается корректным (если, конечно, корректна его контрольная сумма). Если приемник принимает кадр с номером N(S), неравным V(R), то этот кадр отбрасывается и посылается отрицательная квитанция *Отказ* (REJ) с номером V(R). При приеме отрицательной квитанции передатчик обязан повторить передачу кадра с номером V(R), а также всех кадров с большими номерами, которые он уже успел отослать, пользуясь механизмом окна в 127 кадров.

В состав супервизорных кадров входят следующие:

- отказ (REJect);
- приемник не готов (Receiver Not Ready, RNR);
- приемник готов (Receiver Ready, RR).

Команда RR с номером N(R) часто используется как положительная квитанция, когда поток данных от приемника к передатчику отсутствует, а команда RNR – для замедления потока кадров, поступающих на приемник. Это может быть необходимо, если приемник не успевает обработать поток кадров, присылаемых ему с большой скоростью за счет механизма окна. Получение кадра RNR требует от передатчика полной приостановки передачи до получения кадра RR. С помощью этих кадров осуществляется управление потоком данных, что особенно важно для коммутируемых сетей, в которых нет разделяемой среды, автоматически тормозящей работу передатчика за счет того, что новый кадр нельзя передать, пока приемник не закончил прием предыдущего.

2.11. Технология Ethernet (802.3). Метод доступа CSMA/CD

Ethernet – это самый распространенный на сегодняшний день стандарт локальных сетей. Когда говорят *Ethernet*, то под этим обычно понимают любой из вариантов этой технологии. В более узком смысле Ethernet – это сетевой стандарт, основанный на экспериментальной сети Ethernet Network, которую фирма Xerox разработала и реализовала в 1975 году. Метод доступа был опробован еще раньше – во второй половине 60-х годов в радиосети Гавайского университета использовались различные варианты случайного доступа к общей радиосреде, получившие общее название Aloha. В 1980 году 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, но некоторые различия все же имеются. В то время как в стандарте IEEE 802.3 различаются 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 различные модификации – 10Base-5, 10Base-2, 10Base-T, 10Base-FL, 10Base-FB.

В 1995 году был принят стандарт Fast Ethernet, который во многом не является самостоятельным стандартом, о чем говорит и тот факт, что его описание просто является дополнительным разделом к основному стандарту 802.3. Аналогично, принятый в 1998 году стандарт Gigabit Ethernet описан в разделе 802.3z основного документа.

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 (в том числе Fast Ethernet и Gigabit Ethernet) используют один и тот же метод разделения среды передачи данных – метод CSMA/CD.

Метод доступа CSMA/CD

В сетях Ethernet используется метод доступа к среде передачи данных, называемый методом коллективного доступа с опознаванием несущей и обнаружением коллизий (*carrier-sense-multiply-access with collision detection, CSMA/CD*).

Этот метод применяется исключительно в сетях с логической общей шиной (к которым относятся и радиосети, породившие этот метод). Все компьютеры такой сети имеют непосредственный доступ к общей шине, поэтому она может быть использована для передачи данных между любыми двумя узлами сети. Одновременно все компьютеры сети имеют возможность немедленно (с учетом задержки распространения сигнала по физической среде) получить данные, которые любой из компьютеров начал передавать на общую шину (рис. 2.32). Простота схемы подключения – это один из факторов, определивших успех стандарта Ethernet. Говорят, что кабель, к которому подключены все станции, работает в режиме *коллективного доступа* (*Multiply Access, MA*).

Рис. 2.32. Метод случайного доступа CSMA/CD

Этапы доступа к среде

Все данные, передаваемые по сети, помещаются в кадры определенной структуры и снабжаются уникальным адресом станции назначения.

Чтобы получить возможность передавать кадр, станция должна убедиться, что разделяемая среда свободна. Это достигается прослушиванием

основной гармоники сигнала, которая также называется несущей частотой (*carrier-sense, CS*). Признаком незанятости среды является отсутствие на ней несущей частоты, которая при манчестерском способе кодирования равна 5 – 10 МГц, в зависимости от последовательности единиц и нулей, передаваемых в данный момент.

Если среда свободна, то узел имеет право начать передачу кадра. Этот кадр изображен на рис. 2.32 первым. Узел 1 обнаружил, что среда свободна, и начал передавать свой кадр. В классической сети Ethernet на коаксиальном кабеле сигналы передатчика узла 1 распространяются в обе стороны, так что все узлы сети их получают. Кадр данных всегда сопровождается преамбулой (*preamble*), которая состоит из 7 байт, состоящих из значений 10101010, и 8-го байта, равного 10101011. Преамбула нужна для вхождения приемника в побитовый и побайтовый синхронизм с передатчиком.

Все станции, подключенные к кабелю, могут распознать факт передачи кадра, и та станция, которая узнает собственный адрес в заголовках кадра, записывает его содержимое в свой внутренний буфер, обрабатывает полученные данные, передает их вверх по своему стеку, а затем посылает по кабелю кадр-ответ. Адрес станции-источника содержится в исходном кадре, поэтому станция-получатель знает, кому нужно послать ответ.

Узел 2 во время передачи кадра узлом 1 также пытался начать передачу своего кадра, однако обнаружил, что среда занята – на ней присутствует несущая частота, – поэтому узел 2 вынужден ждать, пока узел 1 не прекратит передачу кадра.

После окончания передачи кадра все узлы сети обязаны выдержать технологическую паузу (Inter Packet Gap) в 9,6 мкс. Эта пауза, называемая также межкадровым интервалом, нужна для приведения сетевых адаптеров в исходное состояние, а также для предотвращения монопольного захвата среды одной станцией. После окончания технологической паузы узлы имеют право начать передачу своего кадра, т. к. среда свободна. Из-за задержек распространения сигнала по кабелю не все узлы строго одновременно фиксируют факт окончания передачи кадра узлом 1.

В приведенном примере узел 2 дождался окончания передачи кадра узлом 1, сделал паузу в 9,6 мкс и начал передачу своего кадра.

Возникновение коллизии

При описанном подходе возможна ситуация, когда две станции одновременно пытаются передать кадр данных по общей среде. Механизм прослушивания среды и пауза между кадрами не гарантируют от возник-

новения такой ситуации, когда две или более станции одновременно решают, что среда свободна, и начинают передавать свои кадры. Говорят, что при этом происходит *коллизия* (*collision*), т. к. содержимое обоих кадров сталкивается на общем кабеле и происходит искажение информации – методы кодирования, используемые в Ethernet, не позволяют выделять сигналы каждой станции из общего сигнала.

Коллизия – это нормальная ситуация в работе сетей Ethernet. В примере, изображенном на рис. 2.32, 2.33, коллизию породила одновременная передача данных узлами 1 и 3. Для возникновения коллизии необязательно, чтобы несколько станций начали передачу абсолютно одновременно, такая ситуация маловероятна. Гораздо вероятней, что коллизия возникает из-за того, что один узел начинает передачу раньше другого, но до второго узла сигналы первого просто не успевают дойти к тому времени, когда второй узел решает начать передачу своего кадра. То есть коллизии – это следствие распределенного характера сети.

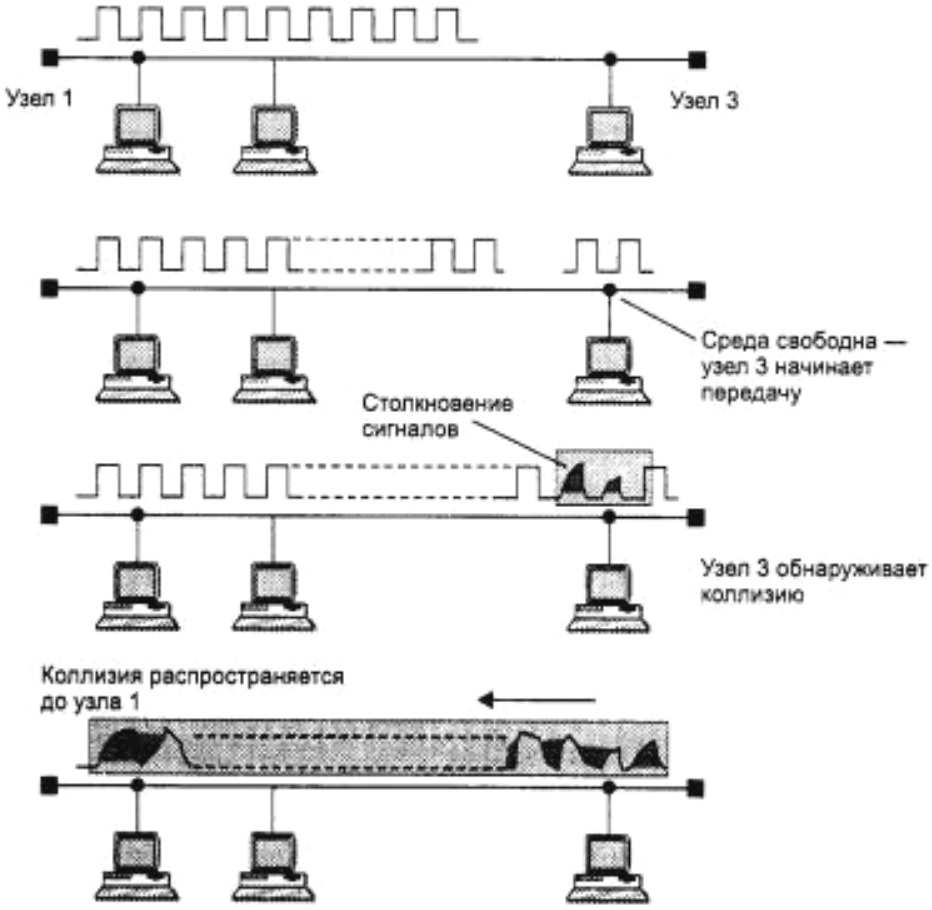


Рис. 2.33. Схема возникновения и распространения коллизии

Чтобы корректно обработать коллизию, все станции одновременно наблюдают за возникающими на кабеле сигналами. Если передаваемые и наблюдаемые сигналы отличаются, то фиксируется *обнаружение коллизии* (*collision detection, CD*). Для увеличения вероятности скорейшего обнаружения коллизии всеми станциями сети станция, которая обнаружила коллизию, прерывает передачу своего кадра (в произвольном месте, возможно, и не на границе байта) и усиливает ситуацию коллизии посылкой в сеть специальной последовательности из 32 бит, называемой *jam-последовательностью.*

После этого обнаружившая коллизию передающая станция обязана прекратить передачу и сделать паузу в течение короткого случайного интервала времени. Затем она может снова предпринять попытку захвата среды и передачи кадра. Случайная пауза выбирается по следующему алгоритму:

$$Пауза = L \cdot (интервал\ отсрочки),$$

где интервал отсрочки равен 512 битовым интервалам (в технологии Ethernet принято все интервалы измерять в битовых интервалах; битовый интервал обозначается как bt и соответствует времени между появлением двух последовательных бит данных на кабеле; для скорости 10 Мбит/с величина битового интервала равна 0,1 мкс или 100 нс); L представляет собой целое число, выбранное с равной вероятностью из диапазона $[0, 2N]$, где N – номер повторной попытки передачи данного кадра: 1, 2, ..., 10.

После 10-й попытки интервал, из которого выбирается пауза, не увеличивается. Таким образом, случайная пауза может принимать значения от 0 до 52,4 мс.

Если 16 последовательных попыток передачи кадра вызывают коллизию, то передатчик должен прекратить попытки и отбросить этот кадр.

Из описания метода доступа видно, что он носит вероятностный характер, и вероятность успешного получения в свое распоряжение общей среды зависит от загруженности сети, т. е. от интенсивности возникновения в станциях потребности в передаче кадров. При разработке этого метода в конце 70-х годов предполагалось, что скорость передачи данных в 10 Мбит/с очень высока по сравнению с потребностями компьютеров во взаимном обмене данными, поэтому загрузка сети будет всегда небольшой. Это предположение остается иногда справедливым и по сей день, однако уже появились приложения, работающие в реальном масштабе времени с мультимедийной информацией, которые очень загружают сегменты Ethernet. При этом коллизии возникают гораздо чаще. При значительной

интенсивности коллизий полезная пропускная способность сети Ethernet резко падает, т. к. сеть почти постоянно занята повторными попытками передачи кадров. Для уменьшения интенсивности возникновения коллизий нужно либо уменьшить трафик, сократив, например, количество узлов в сегменте или заменив приложения, либо повысить скорость протокола, например, перейти на Fast Ethernet.

Следует отметить, что метод доступа CSMA/CD вообще не гарантирует станции, что она когда-либо сможет получить доступ к среде. Конечно, при небольшой загрузке сети вероятность такого события невелика, но при коэффициенте использования сети, приближающемся к 1, такое событие становится очень вероятным. Этот недостаток метода случайного доступа – плата за его чрезвычайную простоту, которая сделала технологию Ethernet самой недорогой. Другие методы доступа – маркерный доступ сетей Token Ring и FDDI, метод Demand Priority сетей 100VG-AnyLAN – свободны от этого недостатка.

Время двойного оборота и распознавание коллизий

Четкое распознавание коллизий всеми станциями сети является необходимым условием корректной работы сети Ethernet. Если какая-либо передающая станция не распознает коллизию и решит, что кадр данных ею передан верно, то этот кадр данных будет утерян. Из-за наложения сигналов при коллизии информация кадра исказится и он будет отбракован принимающей станцией (возможно, из-за несовпадения контрольной суммы). Скорее всего, искаженная информация будет повторно передана каким-либо протоколом верхнего уровня, например транспортным или прикладным, работающим с установлением соединения. Но повторная передача сообщения протоколами верхних уровней произойдет через значительно более длительный интервал времени (иногда даже через несколько секунд) по сравнению с микросекундными интервалами, которыми оперирует протокол Ethernet. Поэтому если коллизии не будут надежно распознаваться узлами сети Ethernet, то это приведет к заметному снижению полезной пропускной способности данной сети.

Для надежного распознавания коллизий должно выполняться следующее соотношение:

$$T_{\min} \geq PDV,$$

где $T_{\min}$ – время передачи кадра минимальной длины, а PDV – время, за которое сигнал коллизии успевает распространиться до самого дальнего узла сети.

Так как в худшем случае сигнал должен пройти дважды между наиболее удаленными друг от друга станциями сети (в одну сторону проходит неискаженный сигнал, а на обратном пути распространяется уже искаженный коллизией сигнал), то это время называется *временем двойного оборота (Path Delay Value, PDV).*

При выполнении этого условия передающая станция должна успевать обнаружить коллизию, которую вызвал переданный ею кадр, еще до того, как она закончит передачу этого кадра.

Очевидно, что выполнение этого условия зависит, с одной стороны, от длины минимального кадра и пропускной способности сети, а с другой стороны – от длины кабельной системы сети и скорости распространения сигнала в кабеле (для разных типов кабеля эта скорость несколько отличается).

Все параметры протокола Ethernet подобраны таким образом, чтобы при нормальной работе узлов сети коллизии всегда четко распознавались. При выборе параметров, конечно, учитывалось и приведенное выше соотношение, связывающее между собой минимальную длину кадра и максимальное расстояние между станциями в сегменте сети.

В стандарте Ethernet принято, что минимальная длина поля данных кадра составляет 46 байт (что вместе со служебными полями дает минимальную длину кадра 64 байт, а вместе с преамбулой – 72 байт или 576 бит). Отсюда может быть определено ограничение на расстояние между станциями.

Итак, в 10-мегабитном Ethernet время передачи кадра минимальной длины равно 575 битовых интервалов, следовательно, время двойного оборота должно быть меньше 57,5 мкс. Расстояние, которое сигнал может пройти за это время, зависит от типа кабеля и для толстого коаксиального кабеля равно примерно 13280 м. Учитывая, что за это время сигнал должен пройти по линии связи дважды, расстояние между двумя узлами не должно быть больше 6635 м. В стандарте величина этого расстояния выбрана существенно меньше, с учетом других, более строгих ограничений.

Одно из таких ограничений связано с предельно допустимым затуханием сигнала. Для обеспечения необходимой мощности сигнала при его прохождении между наиболее удаленными друг от друга станциями сегмента кабеля максимальная длина непрерывного сегмента толстого коаксиального кабеля с учетом вносимого им затухания выбрана в 500 м. Очевидно, что на кабеле в 500 м условия распознавания коллизий будут выполняться с большим запасом для кадров любой стандартной длины, в том числе и 72 байт (время двойного оборота по кабелю 500 м составляет всего

43,3 битовых интервала). Поэтому минимальная длина кадра могла бы быть установлена еще меньше. Однако разработчики технологии не стали уменьшать минимальную длину кадра, имея в виду многосегментные сети, которые строятся из нескольких сегментов, соединенных повторителями.

Повторители увеличивают мощность передаваемых с сегмента на сегмент сигналов, в результате затухание сигналов уменьшается и можно использовать сеть гораздо большей длины, состоящую из нескольких сегментов. В коаксиальных реализациях Ethernet разработчики ограничили максимальное количество сегментов в сети пятью, что, в свою очередь, ограничивает общую длину сети 2500 метрами. Даже в такой многосегментной сети условие обнаружения коллизий по-прежнему выполняется с большим запасом (сравним полученное из условия допустимого затухания расстояние в 2500 м с вычисленным выше максимально возможным по времени распространения сигнала расстоянием 6635 м). Однако в действительности временной запас существенно меньше, поскольку в многосегментных сетях сами повторители вносят в распространение сигнала дополнительную задержку в несколько десятков битовых интервалов. Естественно, небольшой запас был сделан также для компенсации отклонений параметров кабеля и повторителей.

В результате учета всех этих и некоторых других факторов было тщательно подобрано соотношение между минимальной длиной кадра и максимально возможным расстоянием между станциями сети, которое обеспечивает надежное распознавание коллизий. Это расстояние называют также максимальным диаметром сети.

Максимальная производительность сети Ethernet

Количество обрабатываемых кадров Ethernet в секунду часто указывается производителями мостов/коммутаторов и маршрутизаторов как основная характеристика производительности этих устройств. В свою очередь, интересно знать чистую максимальную пропускную способность сегмента Ethernet в кадрах в секунду в идеальном случае, когда в сети нет коллизий и нет дополнительных задержек, вносимых мостами и маршрутизаторами. Такой показатель помогает оценить требования к производительности коммуникационных устройств, т. к. в каждый порт устройства не может поступать больше кадров в единицу времени, чем позволяет это сделать соответствующий протокол.

Для коммуникационного оборудования наиболее тяжелым режимом является обработка кадров минимальной длины. Это объясняется тем, что на обработку каждого кадра мост, коммутатор или маршрутизатор тратит

примерно одно и то же время, связанное с просмотром таблицы продвижения пакета, формированием нового кадра (для маршрутизатора) и т. п. А количество кадров минимальной длины, поступающих на устройство в единицу времени, естественно, больше, чем кадров любой другой длины. Другая характеристика производительности коммуникационного оборудования – бит в секунду – используется реже, т. к. она не говорит о том, какого размера кадры при этом обрабатывало устройство, а на кадрах максимального размера достичь высокой производительности, измеряемой в битах в секунду, гораздо легче.

Рассчитаем максимальную производительность сегмента Ethernet в таких единицах, как число переданных кадров (пакетов) минимальной длины в секунду.

Для расчета максимального количества кадров минимальной длины, проходящих по сегменту Ethernet, заметим, что размер кадра минимальной длины вместе с преамбулой составляет 72 байт или 576 бит (рис. 2.34), поэтому на его передачу затрачивается 57,5 мкс. Прибавив межкадровый интервал в 9,6 мкс, получаем, что период следования кадров минимальной длины составляет 67,1 мкс. Отсюда максимально возможная пропускная способность сегмента Ethernet составляет 14880 кадров в секунду.

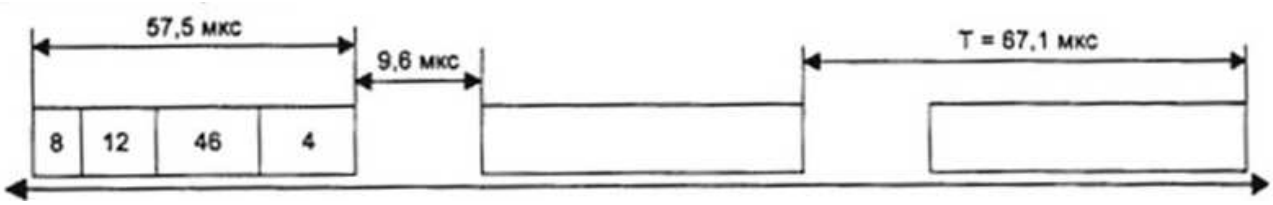


Рис. 2.34. К расчету пропускной способности протокола Ethernet

Естественно, что наличие в сегменте нескольких узлов снижает эту величину за счет ожидания доступа к среде, а также за счет коллизий, приводящих к необходимости повторной передачи кадров.

Кадры максимальной длины технологии Ethernet имеют поле длиной 1500 байт, что вместе со служебной информацией дает 1518 байт, а с преамбулой составляет 1526 байт или 12208 бит. Максимально возможная пропускная способность сегмента Ethernet для кадров максимальной длины составляет 813 кадр/с. Очевидно, что при работе с большими кадрами нагрузка на мосты, коммутаторы и маршрутизаторы довольно ощутимо снижается.

Теперь рассчитаем, какой максимальной полезной пропускной способностью в бит в секунду обладают сегменты Ethernet при использовании кадров разного размера.

Под *полезной пропускной способностью* протокола понимается скорость передачи пользовательских данных, которые переносятся полем данных кадра. Эта пропускная способность всегда меньше номинальной битовой скорости протокола Ethernet за счет нескольких факторов:

– служебной информации кадра;
– межкадровых интервалов (IPG);
– ожидания доступа к среде.

Для кадров минимальной длины полезная пропускная способность равна

$$СП = 14880 \cdot 46 \cdot 8 = 5{,}48\,\text{Мбит/с}.$$

Это намного меньше 10 Мбит/с, но следует учесть, что кадры минимальной длины используются в основном для передачи квитанций, так что к передаче собственно данных файлов эта скорость отношения не имеет.

Для кадров максимальной длины полезная пропускная способность

$$СП = 813 \cdot 1500 \cdot 8 = 9{,}76\,\text{Мбит/с},$$

что весьма близко к номинальной скорости протокола.

Еще раз подчеркнем, что такой скорости можно достигнуть только в том случае, когда двум взаимодействующим узлам в сети Ethernet другие узлы не мешают, что бывает крайне редко.

2.12. Спецификации физической среды Ethernet. Стандарт 10Base-5

Первые сети технологии Ethernet были созданы на коаксиальном кабеле диаметром 0,5 дюйма. В дальнейшем были определены и другие спецификации физического уровня для стандарта Ethernet, позволяющие использовать различные среды передачи данных. Метод доступа CSMA/CD и все временные параметры остаются одними и теми же для любой спецификации физической среды технологии Ethernet 10 Мбит/с.

Физические спецификации технологии Ethernet включают следующие среды передачи данных.

– *l0Base-5* – коаксиальный кабель диаметром 0,5 дюйма, называемый «толстым» коаксиалом. Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента – 500 метров (без повторителей);

– *l0Base-2* – коаксиальный кабель диаметром 0,25 дюйма, называемый «тонким» коаксиалом. Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента – 185 метров (без повторителей);

– *l0Base-T* – кабель на основе неэкранированной витой пары (Unshielded Twisted Pair, UTP). Образует звездообразную топологию на ос-

нове концентратора. Расстояние между концентратором и конечным узлом – не более 100 м;

– *l0Base-F* – волоконно-оптический кабель. Топология аналогична топологии стандарта l0Base-T. Имеется несколько вариантов этой спецификации: FOIRL (расстояние до 1000 м), l0Base-FL (расстояние до 2000 м), l0Base-FB (расстояние до 2000 м).

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.

Стандарт 10Base-5

Стандарт IOBase-5 в основном соответствует экспериментальной сети Ethernet фирмы Xerox и может считаться классическим Ethernet. Он использует в качестве среды передачи данных коаксиальный кабель с волновым сопротивлением 50 Ом, диаметром центрального медного провода 2,17 мм и внешним диаметром около 10 мм («толстый» Ethernet). Такими характеристиками обладают кабели марок RG-SHRG-ll.

Различные компоненты сети, состоящей из трех сегментов, соединенных повторителями, выполненной на толстом коаксиале, показаны на рис. 2.35.

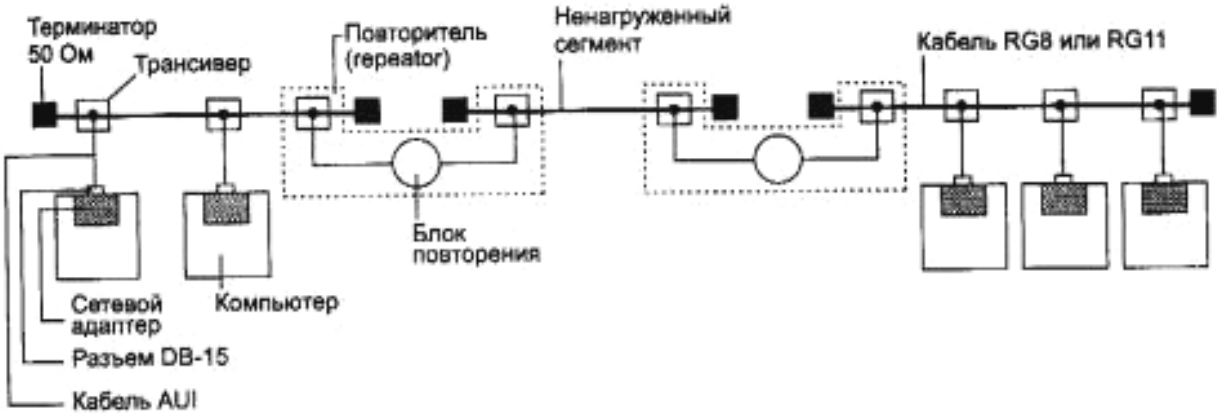


Рис. 2.35. Компоненты физического уровня сети стандарта 10 Base-5, состоящей из трех сегментов

Кабель используется как моноканал для всех станций. Сегмент кабеля имеет максимальную длину 500 м (без повторителей) и должен иметь на концах согласующие терминаторы сопротивлением 50 Ом, поглощающие

распространяющиеся по кабелю сигналы и препятствующие возникновению отраженных сигналов. При отсутствии терминаторов («заглушек») в кабеле возникают стоячие волны, так что одни узлы получают мощные сигналы, а другие – настолько слабые, что их прием становится невозможным.

Станция должна подключаться к кабелю при помощи приемопередатчика – *трансивера* (transmitter + receiver = transceiver). Трансивер устанавливается непосредственно на кабеле и питается от сетевого адаптера компьютера. Трансивер может подсоединяться к кабелю как методом прокалывания, обеспечивающим непосредственный физический контакт, так и бесконтактным методом.

Трансивер соединяется с сетевым адаптером интерфейсным кабелем *AUI (Attachment Unit Interface)* длиной до 50 м, состоящим из 4-х витых пар (адаптер должен иметь разъем AUI). Наличие стандартного интерфейса между трансивером и остальной частью сетевого адаптера очень полезно при переходе с одного типа кабеля на другой. Для этого достаточно только заменить трансивер, а остальная часть сетевого адаптера остается неизменной, т. к. она отрабатывает протокол уровня MAC. При этом необходимо только, чтобы новый трансивер (например, трансивер для витой пары) поддерживал стандартный интерфейс AUI. Для присоединения к интерфейсу AUI используется разъем DB-15.

Допускается подключение к одному сегменту не более 100 трансиверов, причем расстояние между подключениями трансиверов не должно быть меньше 2,5 м. На кабеле имеется разметка через каждые 2,5 м, которая обозначает точки подключения трансиверов. При подсоединении компьютеров в соответствии с разметкой влияние стоячих волн в кабеле на сетевые адаптеры сводится к минимуму.

Трансивер – это часть сетевого адаптера, которая выполняет следующие функции:

- прием и передача данных с кабеля на кабель;
- определение коллизий на кабеле;
- электрическая развязка между кабелем и остальной частью адаптера;
- защита кабеля от некорректной работы адаптера.

Последнюю функцию иногда называют «контролем болтливости», что является буквальным переводом соответствующего английского термина (*jabber control*).

Детектор коллизий определяет наличие коллизии в коаксиальном кабеле по повышенному уровню постоянной составляющей сигналов. Если

постоянная составляющая превышает определенный порог (около 1,5 В), значит, на кабель работает более одного передатчика. Развязывающие элементы (РЭ) обеспечивают гальваническую развязку трансивера от остальной части сетевого адаптера и тем самым защищают адаптер и компьютер от значительных перепадов напряжения, возникающих на кабеле при его повреждении.

Стандарт l0Base-5 определяет возможность использования в сети специального устройства – *повторителя (repeator)*. Повторитель служит для объединения в одну сеть нескольких сегментов кабеля и увеличения тем самым общей длины сети. Повторитель принимает сигналы из одного сегмента кабеля и побитно синхронно повторяет их в другом сегменте, улучшая форму и мощность импульсов, а также синхронизируя импульсы. Повторитель состоит из двух (или нескольких) трансиверов, которые присоединяются к сегментам кабеля, а также блока повторения со своим тактовым генератором. Для лучшей синхроннизации передаваемых бит повторитель задерживает передачу нескольких первых бит преамбулы кадра, за счет чего увеличивается задержка передачи кадра с сегмента на сегмент, а также несколько уменьшается межкадровый интервал IPG.

Стандарт разрешает использование в сети не более 4-х повторителей и, соответственно, не более 5-и сегментов кабеля. При максимальной длине сегмента кабеля в 500 м это дает максимальную длину сети l0Base-5 в 2500 м. Только 3 сегмента из 5-и могут быть нагруженными, т. е. такими, к которым подключаются конечные узлы. Между нагруженными сегментами должны быть ненагруженные сегменты, так что максимальная конфигурация сети представляет собой два нагруженных крайних сегмента, которые соединяются ненагруженными сегментами еще с одним центральным нагруженным сегментом. На рис. 2.35 был приведен пример сети Ethernet, состоящей из трех сегментов, объединенных двумя повторителями. Крайние сегменты являются нагруженными, а промежуточный – ненагруженным.

Правило применения повторителей в сети Ethernet l0Base-5 носит название *«правило 5-4-3»: 5 сегментов, 4 повторителя, 3 нагруженных сегмента*. Ограниченное число повторителей объясняется дополнительными задержками распространения сигнала, которые они вносят. Применение повторителей увеличивает время двойного распространения сигнала, которое для надежного распознавания коллизий не должно превышать время передачи кадра минимальной длины, т. е. кадра в 72 байта или 576 битов.

К *достоинствам* стандарта 10Base-5 относятся:

- хорошая защищенность кабеля от внешних воздействий;
- сравнительно большое расстояние между узлами;

– возможность простого перемещения рабочей станции в пределах длины кабеля AUI.

Недостатками 10Base-5 являются:

– высокая стоимость кабеля;

– сложность его прокладки из-за большой жесткости;

– потребность в специальном инструменте для заделки кабеля;

– остановка работы всей сети при повреждении кабеля или плохом соединении;

– необходимость заранее предусмотреть подводку кабеля ко всем возможным местам установки компьютеров.

2.13. Технология Ethernet. Стандарты 10Base-2, 10Base-T, 10Base-F. Понятие домена коллизий

Стандарт 10Base-2

Стандарт 10Base-2 использует в качестве передающей среды коаксиальный кабель с диаметром центрального медного провода 0,89 мм и внешним диаметром около 5 мм («тонкий» Ethernet). Кабель имеет волновое сопротивление 50 Ом. Такими характеристиками обладают кабели марок RG-58 /U, RG-58 A/U, RG-58 C/U.

Максимальная длина сегмента без повторителей составляет 185 м, сегмент должен иметь на концах согласующие терминаторы 50 Ом. Тонкий коаксиальный кабель дешевле толстого, из-за чего сети l0Base-2 иногда называют сетями Cheapemet (от *cheaper* – более дешевый). Но за дешевизну кабеля приходится расплачиваться качеством – «тонкий» коаксиал обладает худшей помехозащищенностью, худшей механической прочностью и более узкой полосой пропускания.

Станции подключаются к кабелю с помощью высокочастотного BNC Т-коннектора, который представляет собой тройник, один отвод которого соединяется с сетевым адаптером, а два других – с двумя концами разрыва кабеля. Максимальное количество станций, подключаемых к одному сегменту, – 30. Минимальное расстояние между станциями – 1 м. Кабель «тонкого» коаксиала имеет разметку для подключения узлов с шагом в 1 м.

Стандарт l0Base-2 также предусматривает использование повторителей, применение которых также должно соответствовать «правилу 5-4-3». В этом случае сеть будет иметь максимальную длину в $5 \cdot 185 = 925$ м. Очевидно, что это ограничение является более сильным, чем общее ограничение в 2500 метров.

Стандарт 10Base-2 очень близок к стандарту 10Base-5. Но трансиверы в нем объединены с сетевыми адаптерами за счет того, что более гибкий тонкий коаксиальный кабель может быть подведен непосредственно к выходному разъему платы сетевого адаптера, установленной в шасси компьютера. Кабель в данном случае «висит» на сетевом адаптере, что затрудняет физическое перемещение компьютеров.

Типичный состав сети стандарта 10Base-2, состоящей из одного сегмента кабеля, показан на рис. 2.36.

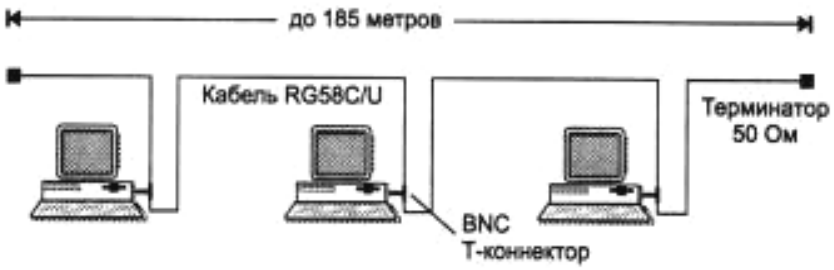


Рис. 2.36. Сеть стандарта 10Base-2

Реализация этого стандарта на практике приводит к наиболее простому решению для кабельной сети, т. к. для соединения компьютеров требуются только сетевые адаптеры, Т-коннекторы и терминаторы 50 Ом. Однако этот вид кабельных соединений наиболее сильно подвержен авариям и сбоям – кабель более восприимчив к помехам, чем «толстый» коаксиал, в моноканале имеется большое количество механических соединений (каждый Т-коннектор дает три механических соединения, два из которых имеют жизненно важное значение для всей сети), пользователи имеют доступ к разъемам и могут нарушить целостность моноканала. Кроме того, эстетика и эргономичность этого решения оставляют желать лучшего, т. к. от каждой станции через Т-коннектор отходят два довольно заметных провода, которые под столом часто образуют моток кабеля – запас, необходимый на случай даже небольшого перемещения рабочего места.

Общим недостатком стандартов 10Base-5 и 10Base-2 является отсутствие оперативной информации о состоянии моноканала. Повреждение кабеля обнаруживается сразу же (сеть перестает работать), но для поиска отказавшего отрезка кабеля необходим специальный прибор – кабельный тестер.

Стандарт 10Base-T

Сети 10Base-T используют в качестве среды две *неэкранированные витые пары* (Unshielded Twisted Pair, UTP). Многопарный кабель на основе неэкранированной витой пары категории 3 (категория определяет поло-

су пропускания кабеля, величину перекрестных наводок NEXT и некоторые другие параметры его качества) телефонные компании уже достаточно давно использовали для подключения телефонных аппаратов внутри зданий. Этот кабель носит также название Voice Grade, говорящее о том, что он предназначен для передачи голоса.

Идея приспособить этот вид кабеля для построения локальных сетей оказалась очень плодотворной, т. к. многие здания уже были оснащены нужной кабельной системой. Оставалось разработать способ подключения сетевых адаптеров и прочего коммуникационного оборудования к витой паре таким образом, чтобы изменения в сетевых адаптерах и программном обеспечении сетевых операционных систем были бы минимальными по сравнению с сетями Ethernet на коаксиале. Поэтому переход на витую пару требует только замены трансивера сетевого адаптера или порта маршрутизатора, а метод доступа и все протоколы канального уровня остались теми же, что и в сетях Ethernet на коаксиале.

Конечные узлы соединяются по топологии «точка – точка» со специальным устройством – многопортовым повторителем – с помощью двух витых пар. Одна витая пара требуется для передачи данных от станции к повторителю (выход Tx сетевого адаптера), а другая – для передачи данных от повторителя к станции (вход Rx сетевого адаптера). На рис. 2.37 показан пример трехпортового повторителя. Повторитель принимает сигналы от одного из конечных узлов и синхронно передает их на все свои остальные порты, кроме того, с которого поступили сигналы.

Многопортовые повторители в данном случае обычно называются концентраторами (англоязычные термины – *hub* или *concentrator*). Концентратор осуществляет функции повторителя сигналов на всех отрезках витых пар, подключенных к его портам, так что образуется единая среда передачи данных – логический моноканал (логическая общая шина). Стандарт определяет битовую скорость передачи данных 10 Мбит/с и максимальное расстояние отрезка витой пары между двумя непосредственно связанными узлами (станциями и концентраторами) не более 100 м при наличии витой пары качества не ниже категории 3. Это расстояние определяется полосой пропускания витой пары – на длине 100 м она позволяет передавать данные со скоростью 10 Мбит/с при использовании манчестерского кода.

Концентраторы 10Base-T можно соединять друг с другом с помощью тех же портов, которые предназначены для подключения конечных узлов. При этом нужно позаботиться о том, чтобы передатчик и приемник одного порта были соединены соответственно с приемником и передатчиком другого порта.

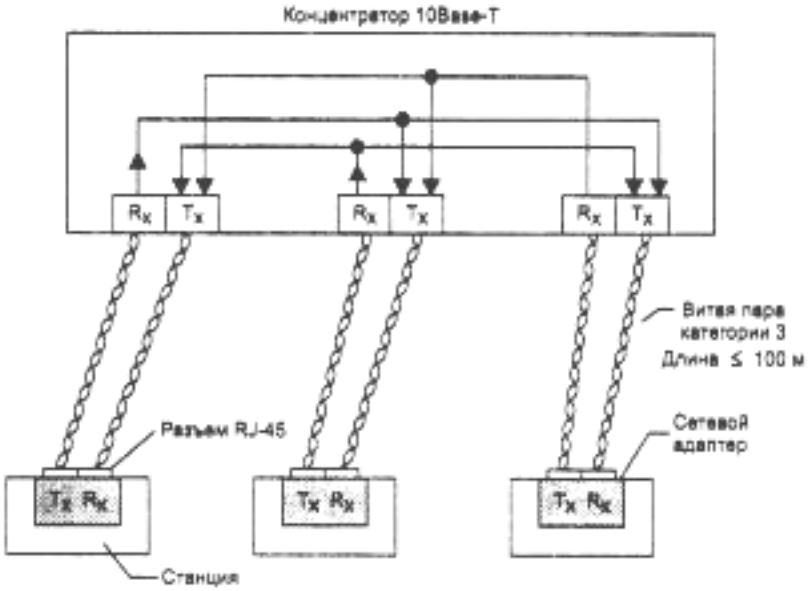


Рис. 2.37. Сеть стандарта 10Base-T: Tx – передатчик; Rx – приемник

Для обеспечения синхронизации станций при реализации процедур доступа CSMA/CD и надежного распознавания станциями коллизий в стандарте определено максимальное число концентраторов между любыми двумя станциями сети – 4. Это правило носит название «правила 4-х хабов» и оно заменяет *«правило 5-4-3»*, применяемое к коаксиальным сетям. При создании сети 10Base-T с большим числом станций концентраторы можно соединять друг с другом иерархическим способом, образуя древовидную структуру (рис. 2.38).

Общее количество станций в сети 10Base-T не должно превышать общего предела в 1024, и для данного типа физического уровня это количество действительно можно достичь. Для этого достаточно создать двухуровневую иерархию концентраторов, расположив на нижнем уровне достаточное количество концентраторов с общим количеством портов 1024 (рис. 2.39). Конечные узлы нужно подключить к портам концентраторов нижнего уровня. Правило 4-х хабов при этом выполняется – между любыми конечными узлами будет ровно 3 концентратора.

Максимальная длина сети в 2500 м здесь понимается как максимальное расстояние между любыми двумя конечными узлами сети (часто применяется также термин «максимальный диаметр сети»). Очевидно, что если между любыми двумя узлами сети не должно быть больше 4-х повторителей, то максимальный диаметр сети 10Base-T составляет $5 \cdot 100 = 500$ м.

Сети, построенные на основе стандарта 10Base-T, обладают по сравнению с коаксиальными вариантами Ethernet многими преимуществами.

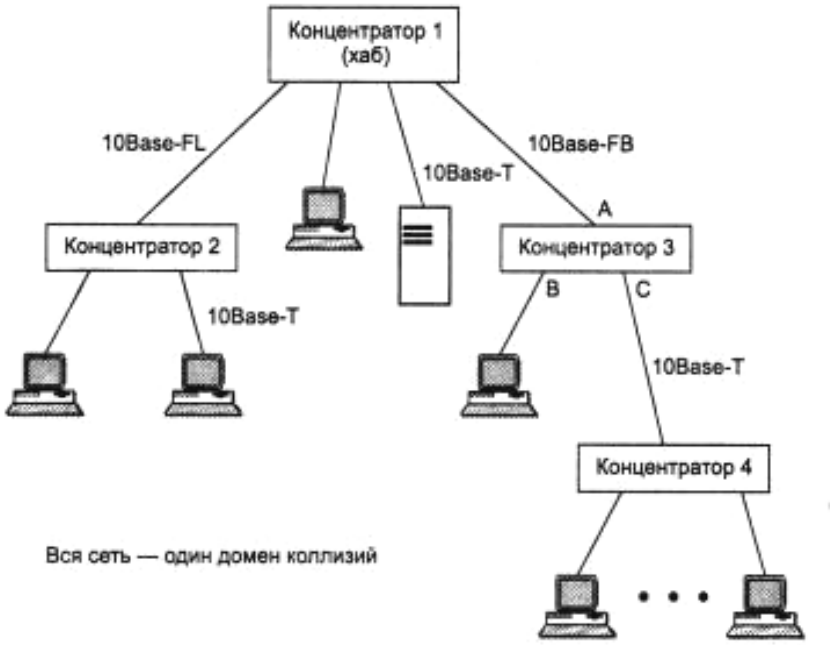


Рис. 2.38. Иерархическое соединение концентраторов Ethernet

Рис. 2.39. Схема с максимальным количеством станций

Эти преимущества связаны с разделением общего физического кабеля на отдельные кабельные отрезки, подключенные к центральному коммуникационному устройству. И хотя логически эти отрезки по-прежнему образуют общую разделяемую среду, их физическое разделение позволяет контролировать их состояние и отключать в случае обрыва, короткого замыкания или неисправности сетевого адаптера на индивидуальной основе. Это обстоятельство существенно облегчает эксплуатацию больших сетей Ethernet, так как концентратор обычно автоматически выполняет такие функции, уведомляя при этом администратора сети о возникшей проблеме.

Стандарт 10Base-F

В качестве среды передачи данных 10-мегабитный Ethernet использует оптическое волокно. Оптоволоконные стандарты в качестве основного типа кабеля рекомендуют достаточно дешевое многомодовое оптическое волокно, обладающее полосой пропускания 500 – 800 МГц при длине кабеля 1 км. Допустимо и более дорогое одномодовое оптическое волокно с полосой пропускания в несколько гигагерц, но при этом нужно применять специальный тип трансивера.

Функционально сеть Ethernet на оптическом кабеле состоит из тех же элементов, что и сеть стандарта 10Base-T: сетевых адаптеров, многопортового повторителя и отрезков кабеля, соединяющих адаптер с портом повторителя. Как и в случае витой пары, для соединения адаптера с повторителем используются два оптоволокна: одно соединяет выход Tx адаптера со входом Rx повторителя, а другое – вход Rx адаптера с выходом Tx повторителя.

Стандарт FOIRL (Fiber Optic Inter-Repeater Link) представляет собой 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 узлами сети – 4. Максимального диаметра в 2500 м здесь достичь можно, хотя максимальные отрезки кабеля между всеми 4-мя повторителями, а также между повторителями и конечными узлами недопустимы, иначе получится сеть длиной 5000 м.

Стандарт 10Base-FL представляет собой незначительное улучшение стандарта FOIRL. Увеличена 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. Конечные узлы не могут 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 они вносят меньшие задержки при передаче данных из одного сегмента в другой, и это является главной причиной, по которой количество повторителей удалось увеличить до 5. В ка-

честве специальных сигналов используются манчестерские коды J и K в следующей последовательности: J-J-K-K-J-J-... Эта последовательность порождает импульсы частотой 2,5 МГц, которые и поддерживают синхронизацию приемника одного концентратора с передатчиком другого. Поэтому стандарт l0Base-FB имеет также название *синхронный Ethernet*.

Как и в стандарте l0Base-T, оптоволоконные стандарты Ethernet разрешают соединять концентраторы только в древовидные иерархические структуры. Любые петли между портами концентраторов не допускаются.

Домен коллизий

В технологии Ethernet, независимо от применяемого стандарта физического уровня, существует понятие домена коллизий.

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.

Приведенная на рис. 2.38 сеть представляет собой один домен коллизий. Если, например, столкновение кадров произошло в концентраторе 4, то в соответствии с логикой работы концентраторов 10Base-T сигнал коллизии распространится по всем портам всех концентраторов.

Если же вместо концентратора 3 поставить в сеть мост, то его порт С, связанный с концентратором 4, воспримет сигнал коллизии, но не передаст его на свои остальные порты, т. к. это не входит в его обязанности. Мост просто отработает ситуацию коллизии средствами порта С, который подключен к общей среде, где эта коллизия возникла. Если коллизия возникла из-за того, что мост пытался передать через порт С кадр в концентратор 4, то, зафиксировав сигнал коллизии, порт С приостановит передачу кадра и попытается передать его повторно через случайный интервал времени. Если порт С принимал в момент возникновения коллизии кадр, то он просто отбросит полученное начало кадра и будет ожидать, когда узел, передававший кадр через концентратор 4, сделает повторную попытку передачи. После успешного принятия данного кадра в свой буфер мост передаст его на другой порт в соответствии с таблицей продвижения, например, на порт А. Все события, связанные с обработкой коллизий портом С, для остальных сегментов сети, которые подключены к другим портам моста, останутся просто неизвестными.

Узлы, образующие один домен коллизий, работают синхронно, как единая распределенная электронная схема.

2.14. Технология Token Ring (802.5)

Основные характеристики технологии

Сети Token Ring, так же как и сети Ethernet, характеризует разделяемая среда передачи данных, которая в данном случае состоит из отрезков кабеля, соединяющих все станции сети в кольцо. Кольцо рассматривается как общий разделяемый ресурс, и для доступа к нему требуется не случайный алгоритм, как в сетях Ethernet, а детерминированный, основанный на передаче станциям права на использование кольца в определенном порядке. Это право передается с помощью кадра специального формата, называемого маркером или токеном (*token*).

Технология Token Ring была разработана компанией IBM в 1984 году, а затем передана в качестве проекта стандарта в комитет IEEE 802, который на ее основе принял в 1985 году стандарт 802.5. Компания IBM использует технологию Token Ring в качестве своей основной сетевой технологии для построения локальных сетей на основе компьютеров различных классов – мэйнфреймов, мини-компьютеров и персональных компьютеров. В настоящее время именно компания IBM является основным законодателем моды технологии Token Ring, производя около 60 % сетевых адаптеров по этой технологии.

Сети Token Ring работают с двумя битовыми скоростями – 4 и 16 Мбит/с. Смешение станций, работающих на различных скоростях, в одном кольце не допускается. Сети Token Ring, работающие со скоростью 16 Мбит/с, имеют некоторые усовершенствования в алгоритме доступа по сравнению со стандартом 4 Мбит/с.

Технология Token Ring – более сложная технология, чем Ethernet. Она обладает свойствами отказоустойчивости. В сети Token Ring определены процедуры контроля работы сети, которые используют обратную связь кольцеобразной структуры – посланный кадр всегда возвращается в станцию-отправитель. В некоторых случаях обнаруженные ошибки в работе сети устраняются автоматически, например, может быть восстановлен потерянный маркер. В других случаях ошибки только фиксируются, а их устранение выполняется вручную обслуживающим персоналом.

Для контроля сети одна из станций выполняет функции так называемого *активного монитора*. Активный монитор выбирается во время инициализации кольца как станция с максимальным значением MAC-адреса. Если активный монитор выходит из строя, процедура инициализации кольца повторяется и выбирается новый активный монитор. Чтобы сеть могла обнаружить отказ активного монитора, последний в работоспособ-

ном состоянии каждые 3 секунды генерирует специальный кадр своего присутствия. Если этот кадр не появляется в сети более 7 секунд, то остальные станции сети начинают процедуру выборов нового активного монитора.

Маркерный метод доступа к разделяемой среде

В сетях с маркерным методом доступа (а к ним, кроме сетей Token Ring, относятся сети FDDI, а также сети, близкие к стандарту 802.4, – ArcNet, сети производственного назначения MAP) право на доступ к среде передается циклически от станции к станции по логическому кольцу.

В сети Token Ring кольцо образуется отрезками кабеля, соединяющими соседние станции. Таким образом, каждая станция связана со своей предшествующей и последующей станцией и может непосредственно обмениваться данными только с ними. Для обеспечения доступа станций к физической среде по кольцу циркулирует кадр специального формата и назначения – маркер. В сети Token Ring любая станция всегда непосредственно получает данные только от одной станции – той, которая является предыдущей в кольце. Такая станция называется ближайшим активным соседом, расположенным выше по потоку (данных) – Nearest Active Upstream Neighbor, NAUN. Передачу же данных станция всегда осуществляет своему ближайшему соседу вниз по потоку данных.

Получив маркер, станция анализирует его и при отсутствии у нее данных для передачи обеспечивает его продвижение к следующей станции. Станция, которая имеет данные для передачи, при получении маркера изымает его из кольца, что дает ей право доступа к физической среде и передачи своих данных. Затем эта станция выдает в кольцо кадр данных установленного формата последовательно по битам. Переданные данные проходят по кольцу всегда в одном направлении от одной станции к другой. Кадр снабжен адресом назначения и адресом источника.

Все станции кольца ретранслируют кадр побитно, как повторители. Если кадр проходит через станцию назначения, то, распознав свой адрес, эта станция копирует кадр в свой внутренний буфер и вставляет в кадр признак подтверждения приема. Станция, выдавшая кадр данных в кольцо, при обратном его получении с подтверждением приема изымает этот кадр из кольца и передает в сеть новый маркер для обеспечения возможности другим станциям сети передавать данные. Такой алгоритм доступа применяется в сетях Token Ring со скоростью работы 4 Мбит/с, описанных в стандарте 802.5.

На рис. 2.40 описанный алгоритм доступа к среде иллюстрируется временно́й диаграммой. Здесь показана передача пакета А в кольце, состоящем из 6-и станций, от станции 1 к станции 3. После прохождения станции назначения 3 в пакете А устанавливаются два признака – признак распознавания адреса и признак копирования пакета в буфер (что на рисунке отмечено звездочкой внутри пакета). После возвращения пакета в станцию 1 отправитель распознает свой пакет по адресу источника и удаляет пакет из кольца. Установленные станцией 3 признаки говорят станции-отправителю о том, что пакет дошел до адресата и был успешно скопирован им в свой буфер.

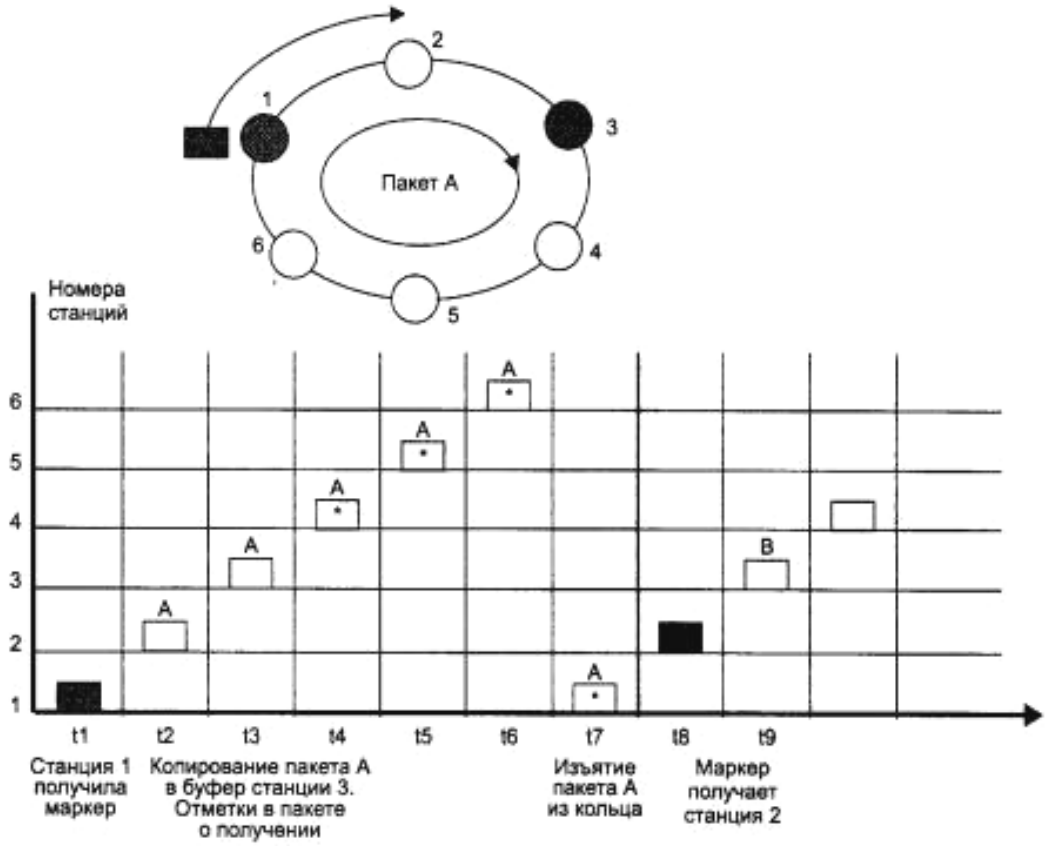


Рис. 2.40. Принцип маркерного доступа

Время владения разделяемой средой в сети Token Ring ограничивается временем удержания маркера (*token holding time*), после истечения которого станция обязана прекратить передачу собственных данных (текущий кадр разрешается завершить) и передать маркер далее по кольцу. Станция может успеть передать за время удержания маркера один или несколько кадров в зависимости от размера кадров и величины времени удержания маркера. Обычно время удержания маркера по умолчанию равно 10 мс, а максимальный размер кадра в стандарте 802.5 не определен.

Для сетей 4 Мбит/с он обычно равен 4 Кбайт, а для сетей 16 Мбит/с – 16 Кбайт. Это связано с тем, что за время удержания маркера станция должна успеть передать хотя бы один кадр.

При скорости 4 Мбит/с за время 10 мс можно передать 5000 байт, а при скорости 16 Мбит/с, соответственно, 20000 байт. Максимальные размеры кадра выбраны с некоторым запасом.

В сетях Token Ring 16 Мбит/с используется также несколько другой 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, т. к. по кольцу одновременно продвигаются кадры нескольких станций. Тем не менее, свои кадры в каждый момент времени может генерировать только одна станция – та, которая в данный момент владеет маркером доступа. Остальные станции в это время только повторяют чужие кадры, так что принцип разделения кольца во времени сохраняется, ускоряется только процедура передачи владения кольцом.

За наличие в сети маркера, причем единственной его копии, отвечает активный монитор. Если активный монитор не получает маркер в течение длительного времени (например, 2,6 с), то он порождает новый маркер.

Приоритетный доступ к кольцу

Для различных видов сообщений, передаваемым кадрам, могут назначаться различные приоритеты: от 0 (низший) до 7 (высший). Решение о приоритете конкретного кадра принимает передающая станция (протокол Token Ring получает этот параметр через межуровневые интерфейсы от протоколов верхнего уровня, например прикладного). Маркер также всегда имеет некоторый уровень текущего приоритета. Станция может воспользоваться маркером, если только у нее есть кадры для передачи с приоритетом, равным или бóльшим, чем приоритет маркера. Сетевой адаптер станции с кадрами, у которых приоритет ниже, чем приоритет маркера, не может захватить маркер, но может поместить наибольший приоритет своих ожидающих передачи кадров в резервные биты маркера, но только в том случае, если записанный в резервных битах приоритет ниже его собственного. В результате в резервных битах приоритета устанавливается наивысший приоритет станции, которая пытается получить доступ к кольцу, но не может этого сделать из-за высокого приоритета маркера.

Станция, сумевшая захватить маркер, передает свои кадры с приоритетом маркера, а затем передает маркер следующему соседу. При этом она переписывает значение резервного приоритета в поле приоритета маркера, а резервный приоритет обнуляется. Поэтому при следующем проходе маркера по кольцу его захватит станция, имеющая наивысший приоритет.

При инициализации кольца основной и резервный приоритеты маркера устанавливаются в 0.

Хотя механизм приоритетов в технологии Token Ring имеется, но он начинает работать только в том случае, когда приложение или прикладной протокол решают его использовать. Иначе все станции будут иметь равные права доступа к кольцу, что в основном и происходит на практике, т. к. бóльшая часть приложений этим механизмом не пользуется. Это связано с тем, что приоритеты кадров поддерживаются не во всех технологиях, например, в сетях Ethernet они отсутствуют, поэтому приложение будет вести себя по-разному, в зависимости от технологии нижнего уровня, что нежелательно. В современных сетях приоритетность обработки кадров обычно обеспечивается коммутаторами или маршрутизаторами.

Физический уровень технологии Token Ring

Стандарт Token Ring фирмы IBM изначально предусматривал построение связей в сети с помощью концентраторов, называемых MAU (Multistation Access Unit) или MSAU (Multi-Station Access Unit), т. е. устройств многостанционного доступа (рис. 2.41). Сеть Token Ring может включать до 260 узлов.

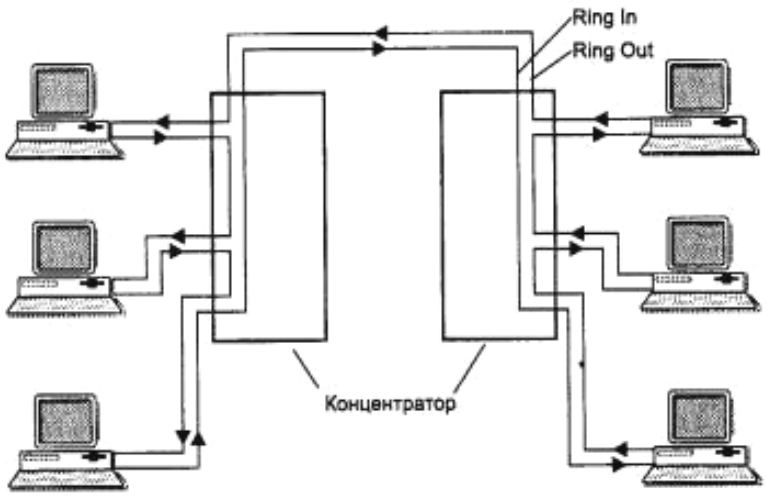


Рис. 2.41. Физическая конфигурация сети Token Ring

Концентратор Token Ring может быть активным или пассивным. Пассивный концентратор просто соединяет порты внутренними связями так, чтобы станции, подключаемые к этим портам, образовали кольцо. Ни усиление сигналов, ни их ресинхронизацию пассивный MSAU не выполняет. Такое устройство можно считать простым кроссовым блоком за одним исключением – MSAU обеспечивает обход какого-либо порта, когда присоединенный к этому порту компьютер выключают. Такая функция необходима для обеспечения связности кольца вне зависимости от состояния подключенных компьютеров. Обычно обход порта выполняется за счет релейных схем, которые питаются постоянным током от сетевого адаптера, а при выключении сетевого адаптера нормально замкнутые контакты реле соединяют вход порта с его выходом.

Активный концентратор выполняет функции регенерации сигналов и поэтому иногда называется повторителем, как в стандарте Ethernet.

Если концентратор является пассивным устройством, то роль усилителя сигналов в этом случае берет на себя каждый сетевой адаптер, а роль ресинхронизирующего блока выполняет сетевой адаптер активного монитора кольца. Каждый сетевой адаптер Token Ring имеет блок повторения, который умеет регенерировать и ресинхронизировать сигналы, однако последнюю функцию выполняет в кольце только блок повторения активного монитора.

Блок ресинхронизации состоит из 30-битного буфера, который принимает манчестерские сигналы с несколько искаженными за время оборота по кольцу интервалами следования. При максимальном количестве станций в кольце (260) вариация задержки циркуляции бита по кольцу может достигать 3-битовых интервалов. Активный монитор «вставляет» свой буфер в кольцо и синхронизирует битовые сигналы, выдавая их на выход с требуемой частотой.

В общем случае сеть Token Ring имеет комбинированную звездно-кольцевую конфигурацию. Конечные узлы подключаются к MSAU по топологии звезды, а сами MSAU объединяются через специальные порты Ring In (RI) и Ring Out (RO) для образования магистрального физического кольца.

Все станции в кольце должны работать на одной скорости – либо 4 Мбит/с, либо 16 Мбит/с. Кабели, соединяющие станцию с концентратором, называются ответвительными (*lobe cable*), а кабели, соединяющие концентраторы, – магистральными (*trunk cable*).

Технология Token Ring позволяет использовать для соединения конечных станций и концентраторов различные типы кабеля – STP Type I, UTP Type 3, UTP Type 6, а также волоконно-оптический кабель.

При использовании экранированной витой пары STP Type 1 из номенклатуры кабельной системы IBM в кольцо допускается объединять до 260 станций при длине ответвительных кабелей до 100 метров, а при использовании неэкранированной витой пары максимальное количество станций сокращается до 72 при длине ответвительных кабелей до 45 метров.

Расстояние между пассивными MSAU может достигать 100 м при использовании кабеля STP Type 1 и 45 м при использовании кабеля UTP Type 3. Между активными MSAU максимальное расстояние увеличивается соответственно до 730 м или 365 м в зависимости от типа кабеля.

Максимальная длина кольца Token Ring составляет 4000 м. Ограничения на максимальную длину кольца и количество станций в кольце в технологии Token Ring не являются такими жесткими, как в технологии Ethernet. Здесь эти ограничения во многом связаны со временем оборота маркера по кольцу (но не только, есть и другие соображения, диктующие выбор ограничений). Так, если кольцо состоит из 260 станций, то при времени удержания маркера в 10 мс маркер вернется в активный монитор в худшем случае через 2,6 с, а это время как раз составляет тайм-аут контроля оборота маркера. В принципе, все значения тайм-аутов в сетевых адаптерах узлов сети Token Ring можно настраивать, поэтому можно построить сеть Token Ring с бóльшим количеством станций и с большей длиной кольца.

Существует большое количество аппаратуры для сетей Token Ring, которая улучшает некоторые стандартные характеристики этих сетей: максимальную длину сети, расстояние между концентраторами, надежность (путем использования двойных колец).

Компания IBM также предложила новый вариант технологии Token Ring, названный High-Speed Token Ring, HSTR. Эта технология поддерживает битовые скорости в 100 и 155 Мбит/с, сохраняя основные особенности технологии Token Ring 16 Мбит/с.

2.15. Технология FDDI

Технология FDDI (Fiber Distributed Data Interface) – оптоволоконный интерфейс распределенных данных – это первая технология локальных сетей, в которой средой передачи данных является волоконно-оптический кабель. Работы по созданию технологий и устройств для использования волоконно-оптических каналов в локальных сетях начались в 80-е годы, вскоре после начала промышленной эксплуатации подобных каналов в территориальных сетях. Проблемная группа X3T9.5 института ANSI раз-

работала в период с 1986 по 1988 годы начальные версии стандарта FDDI, который обеспечивает передачу кадров со скоростью 100 Мбит/с по двойному волоконно-оптическому кольцу длиной до 100 км.

Основные характеристики технологии

Технология FDDI во многом основывается на технологии Token Ring, развивая и совершенствуя ее основные идеи. Разработчики технологии FDDI ставили перед собой в качестве наиболее приоритетных следующие цели:

– повысить битовую скорость передачи данных до 100 Мбит/с;

– повысить отказоустойчивость сети за счет стандартных процедур восстановления ее после отказов различного рода – повреждения кабеля, некорректной работы узла, концентратора, возникновения высокого уровня помех на линии и т. п.;

– максимально эффективно использовать потенциальную пропускную способность сети как для асинхронного, так и для синхронного (чувствительного к задержкам) трафиков.

Сеть FDDI строится на основе двух оптоволоконных колец, которые образуют основной и резервный пути передачи данных между узлами сети. Наличие двух колец – это основной способ повышения отказоустойчивости в сети FDDI, и узлы, которые хотят воспользоваться этим повышенным потенциалом надежности, должны быть подключены к обоим кольцам.

В нормальном режиме работы сети данные проходят через все узлы и все участки кабеля только первичного (Primary) кольца, 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. 2.42), вновь образуя единое кольцо. Этот режим работы сети называется *Wrap*, т. е. *«свертывание»* или *«сворачивание»* колец. Операция свертывания производится средствами концентраторов и/или сетевых адаптеров FDDI. Для упрощения этой процедуры данные по первичному кольцу всегда передаются в одном направлении (на диаграммах это направление изображается против часовой стрелки), а по вторичному – в обратном (изображается по часовой стрелке). Поэтому при образовании общего кольца из двух колец передатчики станций по-прежнему остаются подключенными к приемникам соседних станций, что позволяет правильно передавать и принимать информацию соседними станциями.

Рис. 2.42. Реконфигурация колец FDDI при отказе

В стандартах FDDI много внимания отводится различным процедурам, которые позволяют определить наличие отказа в сети, а затем произвести необходимую реконфигурацию. Сеть FDDI может полностью восстанавливать свою работоспособность в случае единичных отказов ее элементов. При множественных отказах сеть распадается на несколько не связанных сетей. Технология FDDI дополняет механизмы обнаружения отказов технологии Token Ring механизмами реконфигурации пути передачи данных в сети, основанными на наличии резервных связей, обеспечиваемых вторым кольцом.

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 так же называется методом маркерного кольца – *token ring*.

Отличия метода доступа заключаются в том, что время удержания маркера в сети FDDI 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 уменьшаться до нуля. Эти изменения в методе доступа касаются только асинхронного трафика, который не критичен к небольшим задержкам передачи кадров. Для синхронного трафика время удержания маркера по-прежнему остается фиксированной величиной. Механизм приоритетов кадров, аналогичный принятому в технологии Token Ring, в технологии FDDI отсутствует. Разработчики технологии решили, что деление трафика на 8 уровней приоритетов избыточно и достаточно разделить трафик на два класса – асинхронный и синхронный, последний из которых обслуживается всегда, даже при перегрузках кольца.

В остальном пересылка кадров между станциями кольца на уровне MAC полностью соответствует технологии Token Ring. Станции FDDI применяют алгоритм раннего освобождения маркера, как и сети Token Ring со скоростью 16 Мбит/с.

Адреса уровня MAC имеют стандартный для технологий IEEE 802 формат. Формат кадра FDDI близок к формату кадра Token Ring, основные отличия заключаются в отсутствии полей приоритетов. Признаки распознавания адреса, копирования кадра и ошибки позволяют сохранить имеющиеся в сетях Token Ring процедуры обработки кадров станцией-отправителем, промежуточными станциями и станцией-получателем.

Особенности метода доступа FDDI

Для передачи синхронных кадров станция всегда имеет право захватить маркер при его поступлении. При этом время удержания маркера имеет заранее заданную фиксированную величину.

Если же станции кольца FDDI нужно передать асинхронный кадр (тип кадра определяется протоколами верхних уровней), то для выяснения возможности захвата маркера при его очередном поступлении станция должна измерить интервал времени, который прошел с момента предыдущего прихода маркера. Этот интервал называется *временем оборота маркера (Token Rotation Time, TRT)*. Интервал TRT сравнивается с другой величиной – *максимально допустимым временем оборота маркера по кольцу* T_0рг. Если в технологии Token Ring максимально допустимое время оборота маркера является фиксированной величиной (2,6 с из расчета 260 станций в кольце), то в технологии FDDI станции договариваются о величине T_0рг во время инициализации кольца. Каждая станция может предложить свое значение T_0рг, в результате для кольца устанавливается минимальное из предложенных станциями времен. Это позволяет учитывать потребности приложений, работающих на станциях. Обычно синхронным приложениям (приложениям реального времени) нужно чаще передавать данные в сеть небольшими порциями, а асинхронным приложениям лучше получать доступ к сети реже, но большими порциями. Предпочтение отдается станциям, передающим синхронный трафик.

Таким образом, при очередном поступлении маркера для передачи асинхронного кадра сравнивается фактическое время оборота маркера TRT с максимально возможным T_0рг. Если кольцо не перегружено, то маркер

приходит раньше, чем истекает интервал T_0рг, т. е. TRT < T_0рг. В этом случае станции разрешается захватить маркер и передать свой кадр (или кадры) в кольцо. Время удержания маркера THT равно разности T_0рг – TRT, и в течение этого времени станция передает в кольцо столько асинхронных кадров, сколько успеет.

Если же кольцо перегружено и маркер опоздал, то интервал TRT будет больше T_0рг. В этом случае станция не имеет права захватить маркер для асинхронного кадра. Если все станции в сети хотят передавать только асинхронные кадры, а маркер сделал оборот по кольцу слишком медленно, то все станции пропускают маркер в режиме повторения, маркер быстро делает очередной оборот и на следующем цикле работы станции уже имеют право захватить маркер и передать свои кадры.

Метод доступа FDDI для асинхронного трафика является адаптивным и хорошо регулирует временные перегрузки сети.

Отказоустойчивость технологии FDDI

Для обеспечения отказоустойчивости в стандарте FDDI предусмотрено создание двух оптоволоконных колец – первичного и вторичного. В стандарте FDDI допускаются два вида подсоединения станций к сети. Одновременное подключение к первичному и вторичному кольцам называется двойным подключением – Dual Attachment, DA. Подключение только к первичному кольцу называется одиночным подключением – Single Attachment, SA.

В стандарте FDDI предусмотрено наличие в сети конечных узлов – станций (Station), а также концентраторов (Concentrator). Для станций и концентраторов допустим любой вид подключения к сети, как одиночный, так и двойной. Соответственно, такие устройства имеют соответствующие названия: SAS (Single Attachment Station), DAS (Dual Attachment Station), SAC (Single Attachment Concentrator) и DAC (Dual Attachment Concentrator).

Обычно концентраторы имеют двойное подключение, а станции – одинарное, как это показано на рис. 2.43, хотя это и не обязательно. Чтобы устройства легче было правильно присоединять к сети, их разъемы маркируются. Разъемы типа A и B должны быть у устройств с двойным подключением, разъем M (Master) имеется у концентратора для одиночного подключения станции, у которой ответный разъем должен иметь тип S (Slave).

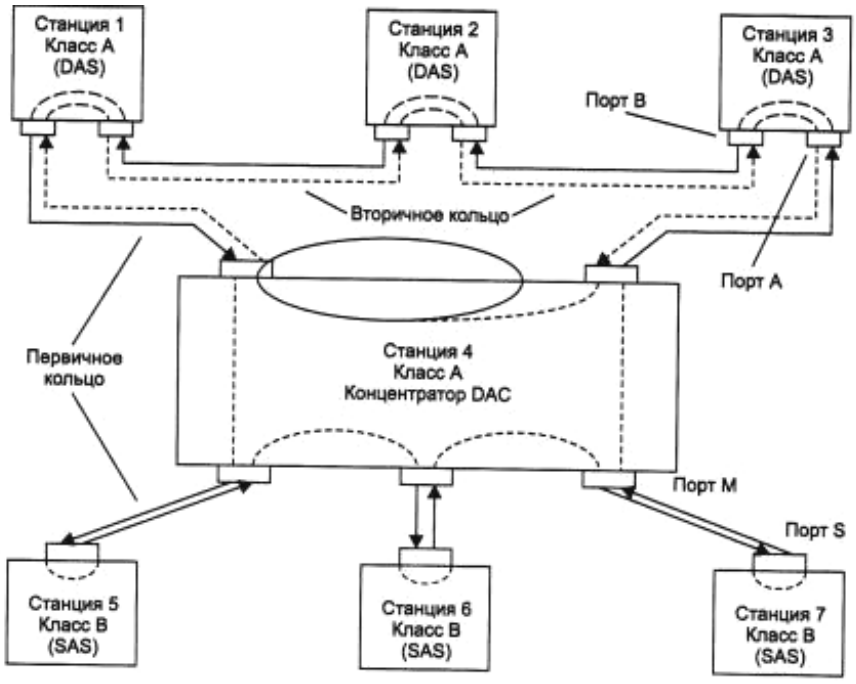


Рис. 2.43. Подключение узлов к кольцам FDDI

В случае однократного обрыва кабеля между устройствами с двойным подключением сеть FDDI сможет продолжить нормальную работу за счет автоматической реконфигурации внутренних путей передачи кадров между портами концентратора (рис. 2.44). Двукратный обрыв кабеля приведет к образованию двух изолированных сетей FDDI. При обрыве кабеля, идущего к станции с одиночным подключением, она становится отрезанной от сети, а кольцо продолжает работать за счет реконфигурации внутреннего пути в концентраторе – порт М, к которому была подключена данная станция, будет исключен из общего пути.

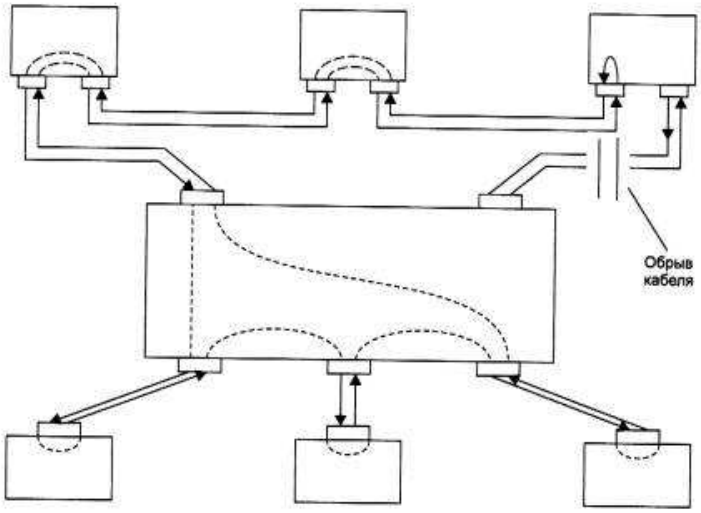


Рис. 2.44. Реконфигурация сети FDDI при обрыве провода

Для сохранения работоспособности сети при отключении питания в станциях с двойным подключением, т. е. станциях DAS, последние должны быть оснащены оптическими обходными переключателями (Optical Bypass Switch), которые создают обходной путь для световых потоков при исчезновении питания, которое они получают от станции.

Станции DAS или концентраторы DAC можно подключать к двум портам M одного или двух концентраторов, создавая древовидную структуру с основными и резервными связями. По умолчанию порт B поддерживает основную связь, а порт A – резервную. Такая конфигурация называется подключением Dual Homing.

Отказоустойчивость поддерживается за счет постоянного слежения уровня SMT концентраторов и станций за временными интервалами циркуляции маркера и кадров, а также за наличием физического соединения между соседними портами в сети. В сети FDDI нет выделенного активного монитора – все станции и концентраторы равноправны, и при обнаружении отклонений от нормы они начинают процесс повторной инициализации сети, а затем и ее реконфигурации.

Реконфигурация внутренних путей в концентраторах и сетевых адаптерах выполняется специальными оптическими переключателями, которые перенаправляют световой луч и имеют достаточно сложную конструкцию.

Физический уровень технологии FDDI

В технологии FDDI для передачи световых сигналов по оптическим волокнам реализовано логическое кодирование 4B/5B в сочетании с физическим кодированием NRZI. Эта схема приводит к передаче по линии связи сигналов с тактовой частотой 125 МГц.

Так как из 32 комбинаций 5-битных символов для кодирования исходных 4-битных символов нужно только 16 комбинаций, то из оставшихся 16 выбрано несколько кодов, которые используются как служебные. К наиболее важным служебным символам относится символ Idle – простой, который постоянно передается между портами в течение пауз между передачей кадров данных. За счет этого станции и концентраторы сети FDDI имеют постоянную информацию о состоянии физических соединений своих портов. В случае отсутствия потока символов Idle фиксируется отказ физической связи и производится реконфигурация внутреннего пути концентратора или станции, если это возможно.

При первоначальном соединении кабелем двух узлов их порты сначала выполняют процедуру установления физического соединения. В этой

процедуре используются последовательности служебных символов кода 4B/5B, с помощью которых создается некоторый язык команд физического уровня. Эти команды позволяют портам выяснить друг у друга типы портов (A, B, M или S) и решить, корректно ли данное соединение (например, соединение S-S является некорректным и т. п.). Если соединение корректно, то далее выполняется тест качества канала при передаче символов кодов 4B/5B, а затем проверяется работоспособность уровня MAC соединенных устройств путем передачи нескольких кадров MAC. Если все тесты прошли успешно, то физическое соединение считается установленным. Работу по установлению физического соединения контролирует протокол управления станцией SMT.

Физический уровень разделен на два подуровня: независимый от среды подуровень PHY (Physical) и зависящий от среды подуровень PMD (Physical Media Dependent).

Технология FDDI в настоящее время поддерживает два подуровня PMD: для волоконно-оптического кабеля и для неэкранированной витой пары категории 5. Последний стандарт появился позже оптического и носит название TP-PMD.

Оптоволоконный подуровень PMD обеспечивает необходимые средства для передачи данных от одной станции к другой по оптическому волокну. Его спецификация определяет:

– использование в качестве основной физической среды многомодового волоконно-оптического кабеля 62,5/125 мкм;

– требования к мощности оптических сигналов и максимальному затуханию между узлами сети. Для стандартного многомодового кабеля эти требования приводят к предельному расстоянию между узлами в 2 км, а для одномодового кабеля расстояние увеличивается до 10 – 40 км в зависимости от качества кабеля;

– требования к оптическим обходным переключателям (*optical bypass switches*) и оптическим приемопередатчикам;

– параметры оптических разъемов MIC (*Media Interface Connector*), их маркировку;

– использование для передачи света с длиной волны в 1300 нм;

– представление сигналов в оптических волокнах в соответствии с методом NRZI.

Подуровень TP-PMD определяет возможность передачи данных между станциями по витой паре в соответствии с методом физического коди-

рования MLT-3, использующего два уровня потенциала: +V и –V для представления данных в кабеле. Для получения равномерного по мощности спектра сигнала данные перед физическим кодированием проходят через скрэмблер. Максимальное расстояние между узлами в соответствии со стандартом TP-PMD равно 100 м.

Максимальная общая длина кольца FDDI составляет 100 километров, максимальное число станций с двойным подключением в кольце – 500.

2.16. Технология Fast Ethernet

Все отличия технологии Fast Ethernet от Ethernet сосредоточены на физическом уровне (рис. 2.45). Уровни MAC и LLC в Fast Ethernet остались абсолютно теми же, и их описывают прежние главы стандартов 802.3 и 802.2. Поэтому, рассматривая технологию Fast Ethernet, мы будем изучать только несколько вариантов ее физического уровня.

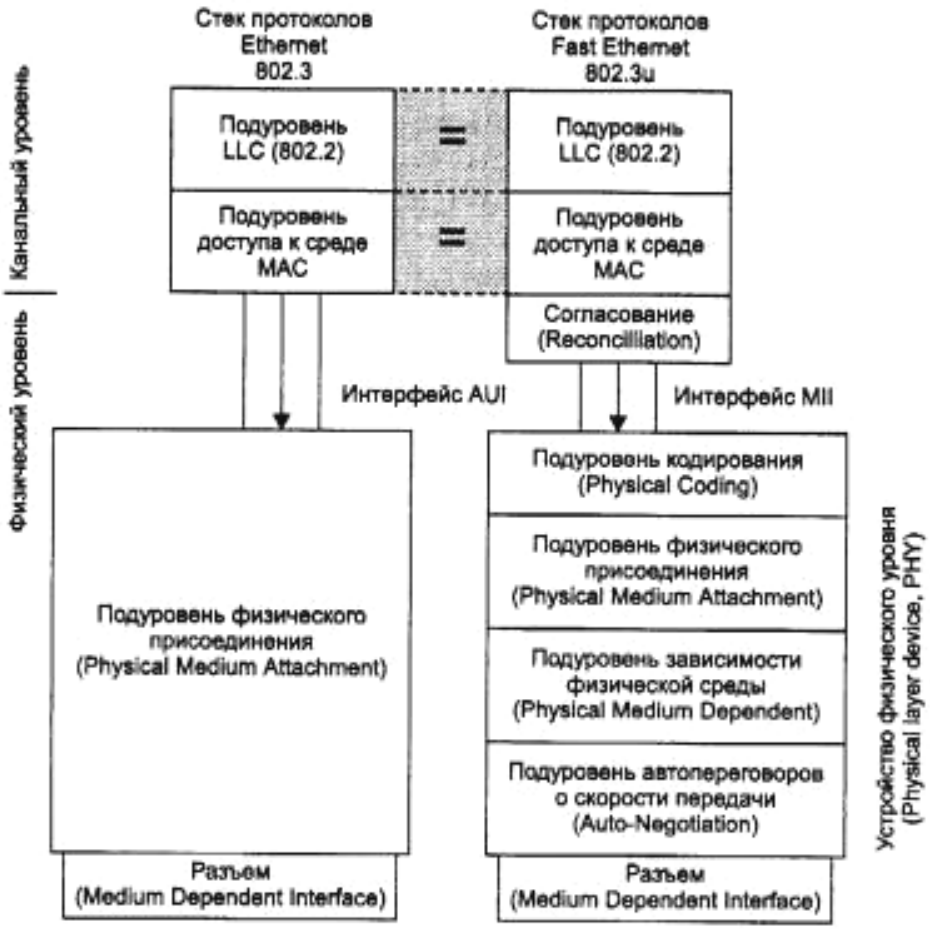


Рис. 2.45. Отличия технологии Fast Ethernet от технологии Ethernet

Более сложная структура физического уровня технологии Fast Ethernet вызвана тем, что в ней используются три варианта кабельных систем:

– волоконно-оптический многомодовый кабель, используются два волокна;

– витая пара категории 5, используются две пары;

– витая пара категории 3, используются четыре пары.

Коаксиальный кабель в число разрешенных сред передачи данных новой технологии Fast Ethernet не попал. Это общая тенденция многих новых технологий, поскольку на небольших расстояниях витая пара категории 5 позволяет передавать данные с той же скоростью, что и коаксиальный кабель, но сеть получается более дешевой и удобной в эксплуатации. На больших расстояниях оптическое волокно обладает гораздо более широкой полосой пропускания, чем коаксиал, а стоимость сети получается ненамного выше, особенно если учесть высокие затраты на поиск и устранение неисправностей в крупной кабельной коаксиальной системе.Отказ от коаксиального кабеля привел к тому, что сети Fast Ethernet всегда имеют иерархическую древовидную структуру, построенную на концентраторах, как и сети l0Base-T/l0Base-F. Основным отличием конфигураций сетей Fast Ethernet является сокращение диаметра сети примерно до 200 м, что объясняется уменьшением времени передачи кадра минимальной длины в 10 раз за счет увеличения скорости передачи в 10 раз по сравнению с 10-мегабитным Ethernet.

Тем не менее, это обстоятельство не очень препятствует построению крупных сетей на технологии Fast Ethernet. Это связано с бурным развитием локальных сетей на основе коммутаторов. При использовании коммутаторов протокол Fast Ethernet может работать в полнодуплексном режиме, в котором нет ограничений на общую длину сети, а остаются только ограничения на длину физических сегментов, соединяющих соседние устройства (адаптер – коммутатор или коммутатор – коммутатор). Поэтому при создании магистралей локальных сетей большой протяженности технология Fast Ethernet также активно применяется, но только в полнодуплексном варианте, совместно с коммутаторами. Остановимся на рассмотрении полудуплексного режима Fast Ethernet.

По сравнению с вариантами физической реализации Ethernet (а их насчитывается шесть), в Fast Ethernet отличия каждого варианта от других глубже – меняется как количество проводников, так и методы кодирования. А так как физические варианты Fast Ethernet создавались одновременно, а не эволюционно, как для сетей Ethernet, то имелась возможность де-

тально определить те подуровни физического уровня, которые не изменяются от варианта к варианту, и те подуровни, которые специфичны для каждого варианта физической среды.

Официальный стандарт 802.3u установил три различных спецификации для физического уровня Fast Ethernet и дал им следующие названия (рис. 2.46):

– 100Base-TX для двухпарного кабеля на неэкранированной витой паре UTP категории 5 или экранированной витой паре STP Type 1;

– 100Base-T4 для четырехпарного кабеля на неэкранированной витой паре UTP категорий 3, 4 или 5;

– 100Base-FX для многомодового оптоволоконного кабеля, используются два волокна.

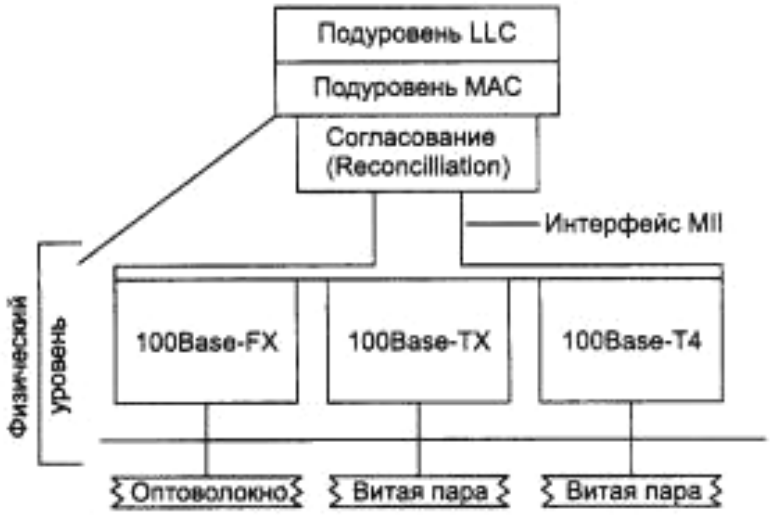


Рис. 2.46. Структура физического уровня Fast Ethernet

Для всех трех стандартов справедливы следующие утверждения и характеристики:

– форматы кадров технологии Fast Ethernetee отличаются от форматов кадров технологий 10-мегабитного Ethernet;

– межкадровый интервал (IPG) равен 0,96 мкс, а битовый интервал равен 10 нс. Все временные параметры алгоритма доступа (интервал отсрочки, время передачи кадра минимальной длины и т. п.), измеренные в битовых интервалах, остались прежними, поэтому изменения в разделы стандарта, касающиеся уровня MAC, не вносились;

– признаком свободного состояния среды является передача по ней символа Idle соответствующего избыточного кода (а не отсутствие сигналов, как в стандартах Ethernet 10 Мбит/с);

– физический уровень включает три элемента:
  а) уровень согласования (*reconciliation sublayer*);
  б) независимый от среды интерфейс (*Media Independent Interface, Mil*);
  в) устройство физического уровня (*Physical layer device, PHY*).

Уровень согласования нужен для того, чтобы уровень MAC, рассчитанный на интерфейс AUI, смог работать с физическим уровнем через интерфейс MII.

Устройство физического уровня (PHY) состоит, в свою очередь, из нескольких подуровней:

– подуровня логического кодирование данных, преобразующего поступающие от уровня MAC байты в символы кода 4B/5B или 8B/6T (оба кода используются в технологии Fast Ethernet);

– подуровней физического присоединения и подуровня зависимости от физической среды (PMD), которые обеспечивают формирование сигналов в соответствии с методом физического кодирования, например NRZI или MLT-3;

– подуровня автопереговоров, который позволяет двум взаимодействующим портам автоматически выбрать наиболее эффективный режим работы, например полудуплексный или полнодуплексный (этот подуровень является факультативным).

Интерфейс MII поддерживает независимый от физической среды способ обмена данными между подуровнем MAC и подуровнем PHY. Этот интерфейс аналогичен по назначению интерфейсу AUI классического 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 МП располагается между подуровнем MAC и подуровнями кодирования сигнала, которых в стандарте Fast Ethernet три: FX, TX и T4.

Разъем MII в отличие от разъема AUI имеет 40 контактов, максимальная длина кабеля MII составляет один метр. Сигналы, передаваемые по интерфейсу МП, имеют амплитуду 5 В.

Физический уровень 100Base-FX – многомодовое оптоволокно, два волокна

Эта спецификация определяет работу протокола Fast Ethernet по многомодовому оптоволокну в полудуплексном и полнодуплексном режимах на основе хорошо проверенной схемы кодирования FDDI. Как и в

стандарте FDDI, каждый узел соединяется с сетью двумя оптическими волокнами, идущими от приемника (Rx) и от передатчика (Tx).

Между спецификациями l00Base-FX и l00Base-TX есть много общего, поэтому общие для двух спецификаций свойства будут даваться под обобщенным названием l00Base-FX/TX.

В то время как Ethernet со скоростью передачи 10 Мбит/с использует манчестерское кодирование для представления данных при передаче по кабелю, в стандарте Fast Ethernet определен другой метод кодирования – 4B/5B. Этот метод уже показал свою эффективность в стандарте FDDI и без изменений перенесен в спецификацию l00Base-FX/TX. При этом методе каждые 4 бита данных подуровня MAC (называемых символами) представляются 5 битами. Избыточный бит позволяет применить потенциальные коды при представлении каждого из пяти битов в виде электрических или оптических импульсов. Существование запрещенных комбинаций символов позволяет отбраковывать ошибочные символы, что повышает устойчивость работы сетей с l00Base-FX/TX.

Для отделения кадра Ethernet от символов Idle используется комбинация символов Start Delimiter (пара символов J (11000) и K (10001) кода 4B/5B, а после завершения кадра перед первым символом Idle вставляется символ T (рис. 2.47).

Рис. 2.47. Непрерывный поток данных спецификаций 100Base-FX/TX

После преобразования 4-битовых порций кодов MAC в 5-битовые порции физического уровня их необходимо представить в виде оптических или электрических сигналов в кабеле, соединяющем узлы сети. Спецификации l00Base-FX и l00Base-TX используют для этого различные методы физического кодирования – NRZI и MLT-3 соответственно (как и в технологии FDDI при работе через оптоволокно и витую пару).

Физический уровень 100Base-TX – витая пара DTP Cat 5 или STP Type 1, две пары

В качестве среды передачи данных спецификация l00Base-TX использует кабель UTP категории 5 или кабель STP Type 1. Максимальная длина кабеля в обоих случаях – 100 м.

Основные отличия от спецификации l00Base-FX – использование метода MLT-3 для передачи сигналов 5-битовых порций кода 4B/5B по витой паре, а также наличие функции *автопереговоров (Auto-negotiation)* для выбора режима работы порта. Схема автопереговоров позволяет двум соединенным физически устройствам, которые поддерживают несколько стандартов физического уровня, отличающихся битовой скоростью и количеством витых пар, выбрать наиболее выгодный режим работы. Обычно процедура автопереговоров происходит при подсоединении сетевого адаптера, который может работать на скоростях 10 и 100 Мбит/с, к концентратору или коммутатору.

Описанная ниже схема Auto-negotiation сегодня является стандартом технологии l00Base-T. До этого производители применяли различные собственные схемы автоматического определения скорости работы взаимодействующих портов, которые не были совместимы. Принятую в качестве стандарта схему Auto-negotiation предложила первоначально компания National Semiconductor под названием NWay.

Всего в настоящее время определено 5 различных режимов работы, которые могут поддерживать устройства l00Base-TX или 100Base-T4 на витых парах:

- l0Base-T – 2 пары категории 3;
- l0Base-T full-duplex – 2 пары категории 3;
- l00Base-TX – 2 пары категории 5 (или Type 1ASTP);
- 100Base-T4 – 4 пары категории 3;
- 100Base-TX full-duplex – 2 пары категории 5 (или Type 1A STP).

Режим l0Base-T имеет самый низкий приоритет при переговорном процессе, а полнодуплексный режим 100Base-T4 – самый высокий. Переговорный процесс происходит при включении питания устройства, а также может быть инициирован в любой момент модулем управления устройства.

Устройство, начавшее процесс auto-negotiation, посылает своему партнеру пачку специальных импульсов *Fast Link Pulse burst (FLP)*, в которой содержится 8-битное слово, кодирующее предлагаемый режим взаимодействия, начиная с самого приоритетного, поддерживаемого данным узлом.

Если узел-партнер поддерживает функцию auto-negotuiation и также может поддерживать предложенный режим, он отвечает пачкой импульсов FLP, в которой подтверждает данный режим, и на этом переговоры заканчиваются. Если же узел-партнер может поддерживать менее приоритетный режим, то он указывает его в ответе, и этот режим выбирается в качестве рабочего. Таким образом, всегда выбирается наиболее приоритетный общий режим узлов.

Узел, который поддерживает только технологию l0Base-T, каждые 16 мс посылает манчестерские импульсы для проверки целостности линии, связывающей его с соседним узлом. Такой узел не понимает запрос FLP, который делает ему узел с функцией Auto-negotiation, и продолжает посылать свои импульсы. Узел, получивший в ответ на запрос FLP только импульсы проверки целостности линии, понимает, что его партнер может работать только по стандарту l0Base-T, и устанавливает этот режим работы и для себя.

Физический уровень 100Base-T4 – витая пара UTP Cat 3, четыре пары

Спецификация 100Base-T4 была разработана для того, чтобы можно было использовать для высокоскоростного Ethernet имеющуюся проводку на витой паре категории 3. Эта спецификация позволяет повысить общую пропускную способность за счет одновременной передачи потоков бит по всем 4-м парам кабеля.

Спецификация 100Base-T4 появилась позже других спецификаций физического уровня Fast Ethernet. Разработчики этой технологии в первую очередь хотели создать физические спецификации, наиболее близкие к спецификациям l0Base-T и l0Base-F, которые работали на двух линиях передачи данных: двух парах или двух волокнах. Для реализации работы по двум витым парам пришлось перейти на более качественный кабель категории 5.

Вместо кодирования 4B/5B в этом методе используется кодирование 8B/6T, которое обладает более узким спектром сигнала и при скорости 33 Мбит/с укладывается в полосу 16 МГц витой пары категории 3 (при кодировании 4B/5B спектр сигнала в эту полосу не укладывается). Каждые 8 бит информации уровня MAC кодируются 6-ю троичными цифрами (ternary symbols), т. е. цифрами, имеющими три состояния. Каждая троичная цифра имеет длительность 40 нс. Группа из 6-ти троичных цифр затем передается на одну из трех передающих витых пар, независимо и последовательно.

Четвертая пара всегда используется для прослушивания несущей частоты в целях обнаружения коллизии. Скорость передачи данных по каждой из трех передающих пар равна 33,3 Мбит/с, поэтому 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.

На рис. 2.48 показано соединение порта MDI сетевого адаптера 100Base-T4 с портом MDI-X концентратора (приставка X говорит о том, что у этого разъема присоединения приемника и передатчика меняются парами кабеля по сравнению с разъемом сетевого адаптера, что позволяет проще соединять пары проводов в кабеле – без перекрещивания). Пара 1-2 всегда требуется для передачи данных от порта MDI к порту MDI-X, пара 3-6 – для приема данных портом MDI от порта MDI-X, а пары 4-5 и 7-8 являются двунаправленными и используются как для приема, так и для передачи, в зависимости от потребности.

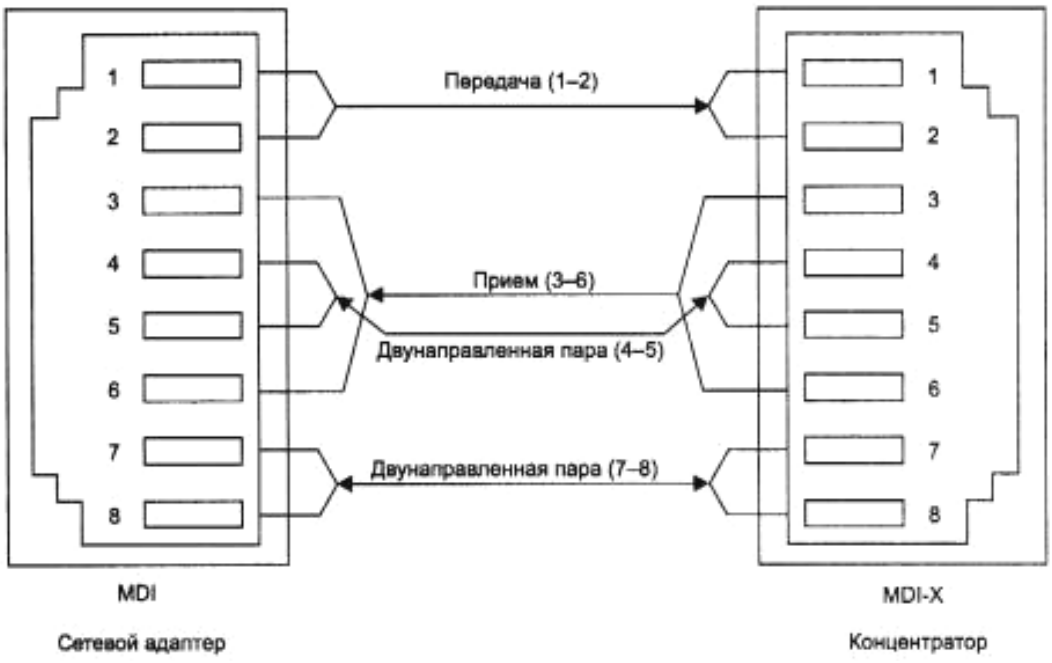


Рис. 2.48. Соединение узлов по спецификации 100Base-T4

Правила построения сегментов Fast Ethernet при использовании повторителей

Технология Fast Ethernet, как и все некоаксиальные варианты Ethernet, рассчитана на использование концентраторов-повторителей для образования связей в сети. 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. Отличительной особенностью DTE является то, что он вырабатывает новый кадр для разделяемого сегмента (мост или коммутатор, хотя и передают через выходной порт кадр, который выработал в свое время сетевой адаптер, но для сегмента сети, к которому подключен выходной порт, этот кадр является новым). Порт повторителя не является DTE, т. к. он побитно повторяет уже появившийся в сегменте кадр.

В типичной конфигурации сети Fast Ethernet несколько DTE подключаются к портам повторителя, образуя сеть звездообразной топологии. Соединения DTE-DTE в разделяемых сегментах не встречаются (если исключить экзотическую конфигурацию, когда сетевые адаптеры двух компьютеров соединены прямо друг с другом кабелем), а вот для мостов/коммутаторов и маршрутизаторов такие соединения являются нормой – когда сетевой адаптер прямо соединен с портом одного из этих устройств либо эти устройства соединяются друг с другом.

Спецификация IEEE 802.3u определяет следующие максимальные длины сегментов DTE-DTE, приведенные в табл. 2.2.

Таблица 2.2

Максимальные длины сегментов DTE-DTE

Стандарт	Тип кабеля	Максимальная длина сегмента
100Base-TX	Категория 5 UTP	100 м
100Base-FX	Многомодовое оптоволокно 62,5/125 мкм	412 м (полудуплекс) 2 км (полный дуплекс)
100Base-T4	Категории 3, 4 или 5 UTP	100 м

Ограничения сетей Fast Ethernet, построенных на повторителях. Повторители Fast Ethernet делятся на два класса. Повторители класса I поддерживают все типы логического кодирования данных, как 4B/5B, так и 8B/6T. Повторители класса II поддерживают только какой-либо один тип логического кодирования – либо 4B/5B, либо 8B/6T. То есть повторители класса I позволяют выполнять трансляцию логических кодов с битовой скоростью 100 Мбит/с, а повторителям класса II эта операция недоступна.

Поэтому повторители класса I могут иметь порты всех трех типов физического уровня: l00Base-TX, l00Base-FX и 100Base-T4. Повторители класса II имеют либо все порты 100Base-T4, либо порты l00Base-TX и l00Base-FX, т. к. последние используют один логический код 4B/5B.

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 – 70 bt.

Повторители класса II вносят меньшую задержку при передаче сигналов: 46 bt для портов TX/FX и 33,5 bt для портов T4. Поэтому максимальное число повторителей класса II в домене коллизий – 2, причем они должны быть соединены между собой кабелем не длиннее 5 метров.

Небольшое количество повторителей Fast Ethernet не является серьезным препятствием при построении больших сетей, так как применение коммутаторов и маршрутизаторов делит сеть на несколько доменов коллизий, каждый из которых будет строиться на одном или двух повторителях. Общая длина сети не будет иметь в этом случае ограничений.

В табл. 2.3 приведены правила построения сети на основе повторителей класса I.

Таблица 2.3

Параметры сетей на основе повторителей класса I

Тип кабелей	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 на оптоволокне	260	100 (TX) 160 (FX)
Несколько сегментов на витой паре и несколько сегментов на оптоволокне	272	100 (TX) 136 (FX)

Эти ограничения проиллюстрированы типовыми конфигурациями сетей, показанными на рис. 2.49.

Таким образом, правило 4-х хабов превратилось для технологии Fast Ethernet в правило одного или двух хабов, в зависимости от класса хаба.

При определении корректности конфигурации сети можно не руководствоваться правилами одного или двух хабов, а рассчитывать время двойного оборота сети, как это было показано выше для сети Ethernet 10 Мбит/с.

Как и для технологии Ethernet 10 Мбит/с, комитет 802.3 дает исходные данные для расчета времени двойного оборота сигнала. Однако при этом сама форма представления этих данных и методика расчета несколько изменились. Комитет предоставляет данные об удвоенных задержках, вносимых каждым элементом сети, не разделяя сегменты сети на левый, пра-

вый и промежуточный. Кроме того, задержки, вносимые сетевыми адаптерами, учитывают преамбулы кадров, поэтому время двойного оборота нужно сравнивать с величиной 512 битовых интервала (bt), то есть со временем передачи кадра минимальной длины без преамбулы.

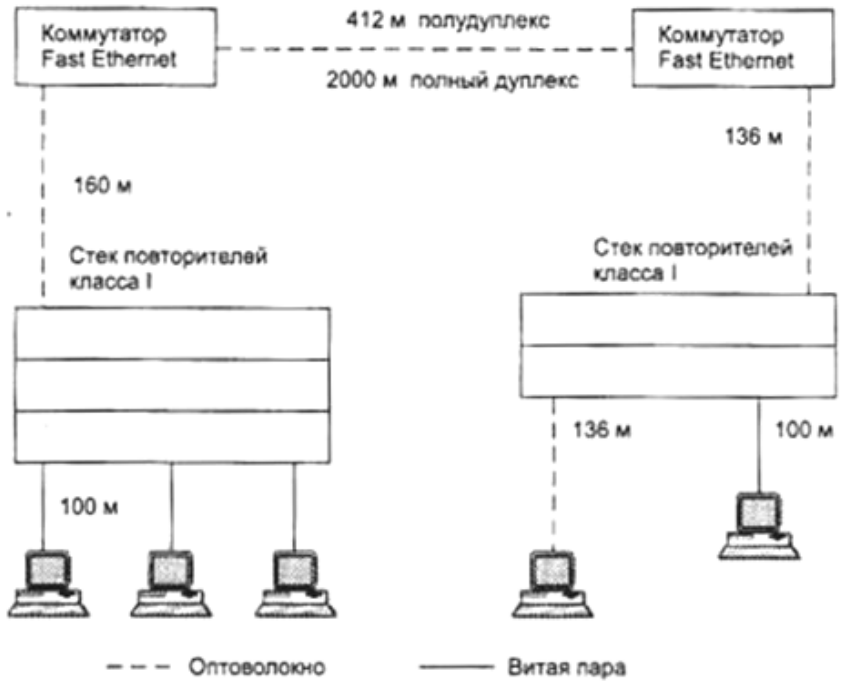


Рис. 2.49. Примеры построения сети Fast Ethernet с помощью повторителей класса I

2.17. Высокоскоростная технология Gigabit Ethernet

Достаточно быстро после появления на рынке продуктов Fast Ethernet сетевые интеграторы и администраторы почувствовали определенные ограничения при построении корпоративных сетей. Во многих случаях серверы, подключенные по 100-мегабитному каналу, перегружали магистрали сетей, работающие также на скорости 100 Мбит/с – магистрали FDDI и Fast Ethernet. Ощущалась потребность в следующем уровне иерархии скоростей.

Летом 1996 года было объявлено о создании группы 802.3z для разработки протокола, максимально подобного Ethernet, но с битовой скоростью 1000 Мбит/с.

Основная идея разработчиков стандарта Gigabit Ethernet состоит в максимальном сохранении идей классической технологии Ethernet при достижении битовой скорости в 1000 Мбит/с. Важно отметить, что Gigabit

Ethernet, так же как и его предшественники, на уровне протокола не должен поддерживать:

– качество обслуживания;

– избыточные связи;

– тестирование работоспособности узлов и оборудования (в последнем случае – за исключением тестирования связи порт – порт, как это делается для Ethernet 10Base-T и 10Base-F и Fast Ethernet).

Главная идея разработчиков технологии Gigabit Ethernet состоит в том, что существуют и будут существовать весьма много сетей, в которых высокая скорость магистрали и возможность назначения пакетам приоритетов в коммутаторах будут вполне достаточны для обеспечения качества транспортного обслуживания всех клиентов сети. И только в тех редких случаях, когда и магистраль достаточно загружена, и требования к качеству обслуживания очень жесткие, нужно применять технологию ATM, которая действительно за счет высокой технической сложности дает гарантии качества обслуживания для всех основных видов трафика.

Избыточные связи и тестирование оборудования не будут поддерживаться технологией Gigabit Ethernet из-за того, что с этими задачами хорошо справляются протоколы более высоких уровней, например Spanning Tree, протоколы маршрутизации и т. п. Поэтому разработчики технологии решили, что нижний уровень просто должен быстро передавать данные, а более сложные и более редко встречающиеся задачи (например, приоритезация трафика) должны передаваться верхним уровням.

Разработка нового стандарта поставила следующие задачи:

- сохранение всех форматов кадров Ethernet;
- поддержка полудуплексной версии протокола на основе метода доступа CSMA/CD и полнодуплексной версии, работающей с коммутаторами;
- поддержка всех основных видов кабелей, используемых в Ethernet и Fast Ethernet (волоконно-оптический, витая пара категории 5, коаксиальный кабель).

Для сохранения приведенных выше свойств разработчикам технологии Gigabit Ethernet пришлось внести изменения не только в физический уровень, как это было в случае Fast Ethernet, но и в уровень MAC.

Основные проблемы, возникшие у разработчиков стандарта:

- задача обеспечения приемлемого диаметра сети для полудуплексного режима работы. В связи с ограничениями, накладываемыми методом CSMA/CD на длину кабеля, версия Gigabit Ethernet для разделяемой среды

допускала бы длину сегмента всего в 25 метров при сохранении размера кадров и всех параметров метода CSMA/CD неизменными;

- достижение битовой скорости 1000 Мбит/с на основных типах кабелей. Даже для оптоволокна достижение такой скорости представляет некоторые проблемы, т. к. технология Fibre Channel, физический уровень которой был взят за основу для оптоволоконной версии Gigabit Ethernet, обеспечивает скорость передачи данных всего в 800 Мбит/с (битовая скорость на линии равна в этом случае примерно 1000 Мбит/с, но при методе кодирования 8B/10B полезная битовая скорость на 25 % меньше скорости импульсов на линии);
- поддержка кабеля на витой паре. Такая задача на первый взгляд кажется неразрешимой, ведь даже для 100-мегабитных протоколов пришлось использовать достаточно сложные методы кодирования, чтобы уложить спектр сигнала в полосу пропускания кабеля.

Средства обеспечения диаметра сети в 200 м на разделяемой среде

Для расширения максимального диаметра сети Gigabit Ethernet в полудуплексном режиме до 200 м были предприняты меры, основывающиеся на соотношении времени передачи кадра минимальной длины и времени двойного оборота.

Минимальный размер кадра был увеличен (без учета преамбулы) с 64 до 512 байт или до 4096 bt. Соответственно, время двойного оборота можно увеличить до 4095 bt, что делает допустимым диаметр сети около 200 м при использовании одного повторителя. При двойной задержке сигнала в 10 bt/m оптоволоконные кабели длиной 100 м вносят вклад во время двойного оборота по 1000 bt, и если повторитель и сетевые адаптеры будут вносить такие же задержки, как в технологии Fast Ethernet, то задержка повторителя в 1000 bt и пары сетевых адаптеров в 1000 bt дадут в сумме время двойного оборота 4000 bt, что удовлетворяет условию распознавания коллизий. Для увеличения длины кадра до требуемой в новой технологии величины сетевой адаптер должен дополнить поле данных до длины 448 байт. Это называется расширением extention, представляющим собой поле, заполненное запрещенными символами кода 8B/10B, которые невозможно принять за коды данных.

Для сокращения накладных расходов при использовании слишком длинных кадров для передачи коротких квитанций конечным узлам разрешено передавать несколько кадров подряд, без передачи среды другим станциям. Такой режим получил название Burst Mode – монопольный пакетный режим. Станция может передать подряд несколько кадров с общей

длиной не более 65536 бит или 8192 байт. Если станции нужно передать несколько небольших кадров, то она может не дополнять их до размера в 512 байт, а передавать подряд до исчерпания предела в 8192 байт (в этот предел входят все байты кадра, в том числе преамбула, заголовок, данные и контрольная сумма). Предел 8192 байт называется BurstLength. Если станция начала передавать кадр и предел BurstLength был достигнут в середине кадра, то кадр разрешается передать до конца.

Увеличение «совмещенного» кадра до 8192 байт несколько задерживает доступ к разделяемой среде других станций, но при скорости 1000 Мбит/с эта задержка не столь существенна.

Спецификации физической среды стандарта 802.3z

В стандарте 802.3z определены следующие типы физической среды:

- одномодовый волоконно-оптический кабель;
- многомодовый волоконно-оптический кабель 62,5/125;
- многомодовый волоконно-оптический кабель 50/125;
- двойной коаксиал с волновым сопротивлением 75 Ом.

Многомодовый кабель

Для передачи данных по традиционному для компьютерных сетей многомодовому волоконно-оптическому кабелю стандарт определяет применение излучателей, работающих на двух длинах волн – 1300 и 850 нм. Применение светодиодов с длиной волны 850 нм объясняется тем, что они намного дешевле, чем светодиоды, работающие на волне 1300 нм, хотя при этом максимальная длина кабеля уменьшается, т. к. затухание многомодового оптоволокна на волне 850 м более чем в два раза выше, чем на волне 1300 нм. Однако возможность удешевления чрезвычайно важна для такой в целом дорогой технологии, как Gigabit Ethernet.

Для многомодового оптоволокна стандарт 802.3z определил спецификации l000Base-SX и l000Base-LX.

В первом случае используется длина волны 850 нм (S означает *Short Wavelength* – короткая волна), а во втором – 1300 нм (L – от *Long Wavelength* – длинная волна).

Для спецификации l000Base-SX предельная длина оптоволоконного сегмента для кабеля 62,5/125 составляет 220 м, а для кабеля 50/125 – 500 м. Эти максимальные значения могут достигаться только для полнодуплексной передачи данных, т. к. время двойного оборота сигнала на двух отрезках 220 м равно 4400 bt, что превосходит предел 4095 bt даже без учета повторителя и сетевых адаптеров. Для полудуплексной передачи максимальные значения сегментов оптоволоконного кабеля всегда должны

быть меньше 100 м. Приведенные расстояния в 220 и 500 м рассчитаны для худшего по стандарту случая полосы пропускания многомодового кабеля, находящегося в пределах от 160 до 500 МГц/км. Реальные кабели обычно обладают значительно лучшими характеристиками, находящимися между 600 и 1000 МГц/км. В этом случае можно увеличить длину кабеля до примерно 800 м.

Одномодовый кабель

Для спецификации l000Base-LX в качестве источника излучения всегда применяется полупроводниковый лазер с длиной волны 1300 нм.

Основная область применения стандарта l000Base-LX – это одномодовое оптоволокно. Максимальная длина кабеля для одномодового волокна равна 5000 м.

Спецификация l000Base-LX может работать и на многомодовом кабеле. В этом случае предельное расстояние получается небольшим – 550 м. Это связано с особенностями распространения когерентного света в широком канале многомодового кабеля. Для присоединения лазерного трансивера к многомодовому кабелю необходимо использовать специальный адаптер.

Твинаксиальный кабель

В качестве среды передачи данных используется высококачественный твинаксиальный кабель (Twinax) с волновым сопротивлением 150 Ом (2×75 Ом). Данные посылаются одновременно по паре проводников, каждый из которых окружен экранирующей оплеткой. При этом получается режим полудуплексной передачи. Для обеспечения полнодуплексной передачи необходимы еще две пары коаксиальных проводников. Начал выпускаться специальный кабель, который содержит четыре коаксиальных проводника – так называемый Quad-кабель. Он внешне напоминает кабель категории 5 и имеет близкий к нему внешний диаметр и гибкость. Максимальная длина твинаксиального сегмента составляет всего 25 метров, поэтому это решение подходит для оборудования, расположенного в одной комнате.

Gigabit Ethernet на витой паре категории 5 (802.3ab)

Каждая пара кабеля категории 5 имеет гарантированную полосу пропускания до 100 МГц. Для передачи по такому кабелю данных со скоростью 1000 Мбит/с было решено организовать параллельную передачу одновременно по всем 4-м парам кабеля (так же, как и в технологии l00VG-AnyLAN). Это уменьшило скорость передачи данных по каждой паре до 250 Мбит/с. Однако и для такой скорости необходимо было реализовать

метод кодирования, который имел бы спектр не выше 100 МГц. Кроме того, одновременное использование четырех пар на первый взгляд лишает сеть возможности распознавать коллизии.

Для кодирования данных был применен код PAM5, использующий 5 уровней потенциала: –2, –1, 0, +1, +2. За один такт по одной паре передается 2,322 бит информации. Следовательно, тактовую частоту вместо 250 МГц стало возможным снизить до 125 МГц. При этом если использовать не все коды, а передавать 8 бит за такт (по 4-м парам), то выдерживается требуемая скорость передачи в 1000 Мбит/с и еще остается запас неиспользуемых кодов, т. к. код PAM5 содержит $5^4 = 625$ комбинаций, а если передавать за один такт по всем четырем парам 8 бит данных, то для этого требуется всего $2^8 = 256$ комбинаций. Оставшиеся комбинации приемник может использовать для контроля принимаемой информации и выделения правильных комбинаций на фоне шума. Код PAM5 на тактовой частоте 125 МГц укладывается в полосу 100 МГц кабеля категории 5.

Для распознавания коллизий и организации полнодуплексного режима разработчики спецификации 802.3aЬ применили технику, используемую при организации дуплексного режима на одной паре проводов в современных модемах и аппаратуре передачи данных абонентских окончаний ISDN. Вместо передачи по разным парам проводов или разнесения сигналов двух одновременно работающих навстречу передатчиков по диапазону частот оба передатчика работают навстречу друг другу по каждой из 4-х пар в одном и том же диапазоне частот, т. к. используют один и тот же потенциальный код PAM5 (рис. 2.50).

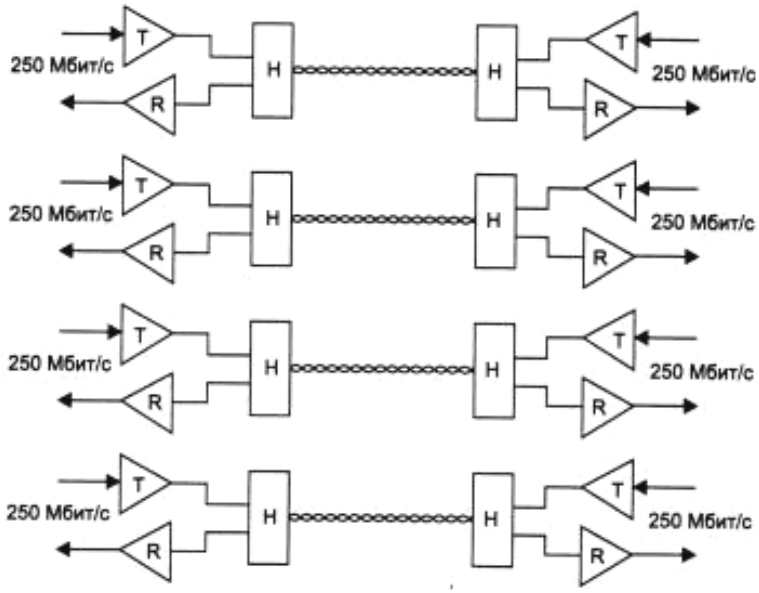


Рис. 2.50. Двунаправленная передача по четырем парам DTP категории 5

Схема гибридной развязки Н позволяет приемнику и передатчику одного и того же узла использовать одновременно витую пару и для приема, и для передачи (так же, как и в трансиверах коаксиального Ethernet).

Для отделения принимаемого сигнала от своего собственного приемник вычитает из результирующего сигнала известный ему свой сигнал. Для выполнения этой операции используются специальные цифровые сигнальные процессоры – DSP (Digital Signal Processor).

При полудуплексном режиме работы получение встречного потока данных считается коллизией, а для полнодуплексного режима работы – нормальной ситуацией.

2.18. Структурированная кабельная система

Кабельная система является фундаментом любой сети. Если в кабелях ежедневно происходят короткие замыкания, нарушаются контакты разъемов, то добавление новой станции приводит к необходимости тестирования десятков контактов разъемов из-за того, что документация на физические соединения не ведется. Становится ясно, что на основе такой кабельной системы любое, самое современное и производительное оборудование будет работать плохо. Пользователи будут недовольны большими периодами простоев и низкой производительностью сети, а обслуживающий персонал будет постоянно разыскивать места коротких замыканий, обрывов и плохих контактов. Причем проблем с кабельной системой становится намного больше при увеличении размеров сети.

Ответом на высокие требования к качеству кабельной системы стали структурированные кабельные системы.

Иерархия в кабельной системе

Структурированная кабельная система (Structured Cabling System, SCS) – это набор коммутационных элементов (кабелей, разъемов, коннекторов, кроссовых панелей и шкафов), а также методика их совместного использования, которая позволяет создавать регулярные, легко расширяемые структуры связей в вычислительных сетях.

Структурированная кабельная система представляет своего рода «конструктор», с помощью которого проектировщик сети строит нужную ему конфигурацию из стандартных кабелей, соединенных стандартными разъемами и коммутируемых на стандартных кроссовых панелях. При необходимости конфигурацию связей можно легко изменить – добавить

компьютер, сегмент, коммутатор, изъять ненужное оборудование, а также поменять соединения между компьютерами и концентраторами.

При построении структурированной кабельной системы подразумевается, что каждое рабочее место на предприятии должно быть оснащено розетками для подключения телефона и компьютера, даже если в данный момент этого не требуется. То есть хорошая структурированная кабельная система строится избыточной. В будущем это может сэкономить средства, т. к. изменения в подключении новых устройств можно производить за счет перекоммутации уже проложенных кабелей.

Структурированная кабельная система планируется и строится иерархически, с главной магистралью и многочисленными ответвлениями от нее (рис. 2.51).

Такая система может быть построена на базе уже существующих современных телефонных кабельных систем, в которых кабели, представляющие собой набор витых пар, прокладываются в каждом здании, разводятся между этажами, на каждом этаже используется специальный кроссовый шкаф, от которого провода в трубах и коробах подводятся к каждой комнате и разводятся по розеткам.

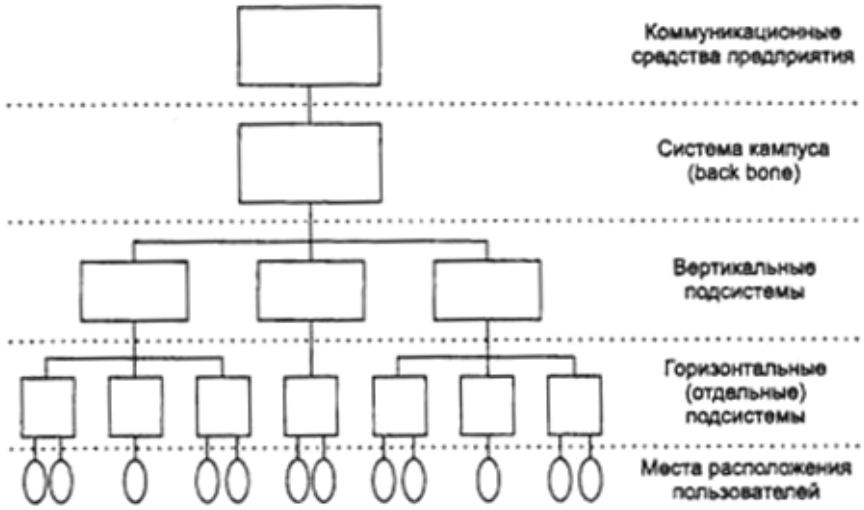


Рис. 2.51. Иерархия структурированной кабельной системы

Типичная иерархическая структура структурированной кабельной системы включает:

– горизонтальные подсистемы (в пределах этажа);

– вертикальные подсистемы (внутри здания);

– подсистему кампуса (в пределах одной территории с несколькими зданиями).

Горизонтальная подсистема соединяет кроссовый шкаф этажа с розетками пользователей. Подсистемы этого типа соответствуют этажам здания. *Вертикальная подсистема* соединяет кроссовые шкафы каждого этажа с центральной аппаратной здания. *Подсистема кампуса* соединяет несколько зданий с главной аппаратной всего кампуса. Эта часть кабельной системы обычно называется *магистралью* (*backbone*).

Использование структурированной кабельной системы вместо хаотически проложенных кабелей дает предприятию много преимуществ:

– *универсальность*. Структурированная кабельная система при продуманной организации может стать единой средой для передачи компьютерных данных в локальной вычислительной сети, организации локальной телефонной сети, передачи видеоинформации и даже передачи сигналов от датчиков пожарной безопасности или охранных систем. Это позволяет автоматизировать многие процессы контроля, мониторинга и управления хозяйственными службами и системами жизнеобеспечения предприятия;

– *увеличение срока службы*. Срок морального старения хорошо структурированной кабельной системы может составлять 10 – 15 лет;

– *уменьшение стоимости добавления новых пользователей* и изменения их мест размещения. Известно, что стоимость кабельной системы значительна и определяется в основном не стоимостью кабеля, а стоимостью работ по его прокладке. Поэтому более выгодно провести однократную работу по прокладке кабеля, возможно, с бóльшим запасом по длине, чем несколько раз выполнять прокладку, наращивая длину кабеля. При таком подходе все работы по добавлению или перемещению пользователя сводятся к подключению компьютера к уже имеющейся розетке;

– *возможность легкого расширения сети*. Структурированная кабельная система – модульная, поэтому ее легко расширять. Например, к магистрали можно добавить новую подсеть, не оказывая никакого влияния на существующие подсети. Можно заменить в отдельной подсети тип кабеля независимо от остальной части сети. Структурированная кабельная система является основой для деления сети на легкоуправляемые логические сегменты, т. к. она сама уже разделена на физические сегменты;

– *обеспечение более эффективного обслуживания*. Структурированная кабельная система облегчает обслуживание и поиск неисправностей по сравнению с шинной кабельной системой. При шинной организации кабельной системы отказ одного из устройств или соединительных элементов приводит к трудно локализуемому отказу всей сети. В структурированных кабельных системах отказ одного сегмента не действует на

другие, т. к. объединение сегментов осуществляется с помощью концентраторов. Концентраторы диагностируют и локализуют неисправный участок;

– *надежность*. Структурированная кабельная система имеет повышенную надежность, поскольку производитель такой системы гарантирует не только качество ее отдельных компонентов, но и их совместимость.

Выбор типа кабеля для горизонтальных подсистем

Разработка структурированной кабельной системы чаще всего начинается с горизонтальных подсистем, т. к. именно к ним подключаются конечные пользователи. При этом выбор типа кабеля, как правило, лежит между экранированной витой парой, неэкранированной витой парой, коаксиальным кабелем и волоконно-оптическим кабелем. Возможно использование и беспроводных линий связи.

Горизонтальная подсистема характеризуется очень большим количеством ответвлений кабеля (рис. 2.52), т. к. его нужно провести к каждой пользовательской розетке, причем и в тех комнатах, где пока компьютеры в сеть не объединяются.

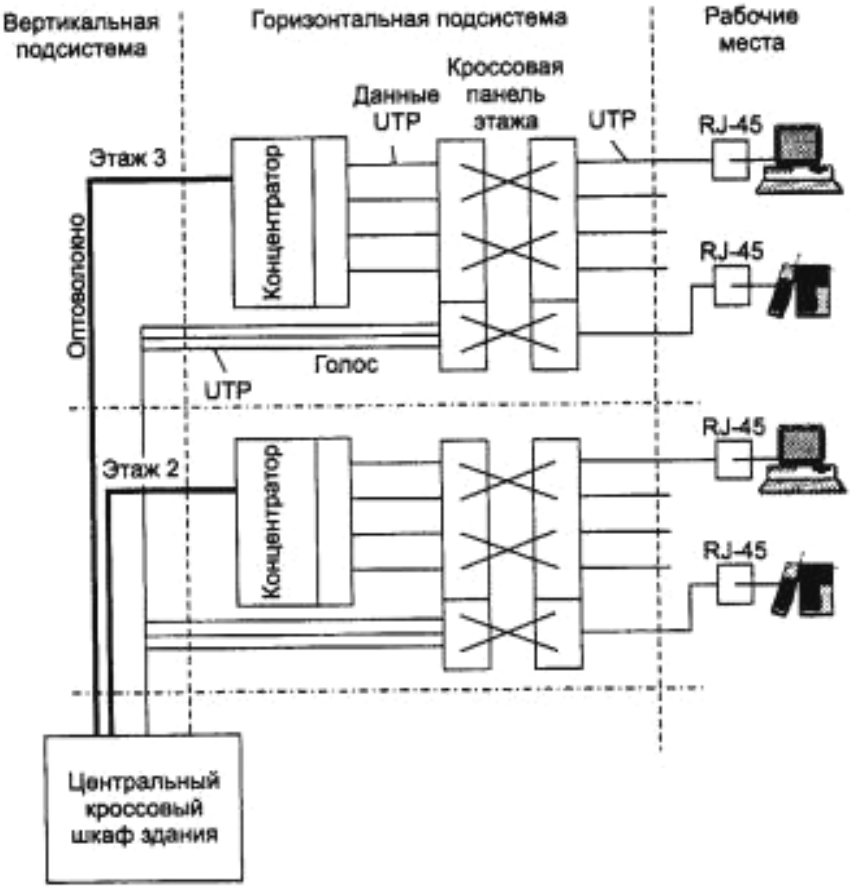


Рис. 2.52. Структура кабельной системы этажа и здания

Поэтому к кабелю, используемому в горизонтальной проводке, предъявляются повышенные требования к удобству выполнения ответвлений, а также удобству его прокладки в помещениях. На этаже обычно устанавливается кроссовая панель, которая позволяет с помощью коротких отрезков кабеля, оснащенного разъемами, провести перекоммутацию соединений между пользовательским оборудованием и концентраторами/коммутаторами.

Медный провод, в частности неэкранированная витая пара, является предпочтительной средой для горизонтальной кабельной подсистемы, хотя, если пользователям нужна очень высокая пропускная способность или кабельная система прокладывается в агрессивной среде, для нее лучше подойдет волоконно-оптический кабель. Коаксиальный кабель – это устаревшая технология, которой следует избегать, если только она уже широко не используется на предприятии. Беспроводная связь является новой и многообещающей технологией, однако из-за сравнительной новизны и низкой помехоустойчивости лучше ограничивать масштабы ее использования неответственными областями.

При выборе кабеля принимаются во внимание следующие характеристики: полоса пропускания, расстояние, физическая защищенность, электромагнитная помехозащищенность, стоимость. Кроме того, при выборе кабеля нужно учитывать, какая кабельная система уже установлена на предприятии, а также какие тенденции и перспективы существуют на рынке в данный момент.

Неэкранированная витая пара UTP по характеристикам полосы пропускания и поддерживаемым расстояниям подходит для создания горизонтальных подсистем.

Однако и коаксиальный кабель все еще остается одним из возможных вариантов кабеля для горизонтальных подсистем, особенно в случаях, когда высокий уровень электромагнитных помех не позволяет использовать витую пару или же небольшие размеры сети не создают больших проблем с эксплуатацией кабельной системы.

Толстый Ethernet обладает по сравнению с тонким большей полосой пропускания, он более стоек к повреждениям и передает данные на бóльшие расстояния, однако к нему сложнее подсоединиться и он менее гибок. С толстым Ethernet сложнее работать, и он мало подходит для горизонтальных подсистем. Однако его можно использовать в вертикальной подсистеме в качестве магистрали, если оптоволоконный кабель по каким-то причинам не подходит.

Тонкий Ethernet – это кабель, который должен был решить проблемы, связанные с применением толстого Ethernet. До появления стандарта 10Base-T тонкий Ethernet был основным кабелем для горизонтальных подсистем. Тонкий Ethernet проще монтировать, чем толстый. Сети на тонком Ethernet можно быстро собрать, т. к. компьютеры соединяются друг с другом непосредственно.

Главный недостаток тонкого Ethernet – сложность его обслуживания. Каждый конец кабеля должен завершаться терминатором 50 Ом. При отсутствии терминатора или утере им своих рабочих свойств (например, из-за отсутствия контакта) перестает работать весь сегмент сети, подключенный к этому кабелю. Аналогичные последствия имеет плохое соединение любой рабочей станции (осуществляемое через Т-коннектор). Неисправности в сетях на тонком Ethernet сложно локализовать. Часто приходится отсоединять Т-коннектор от сетевого адаптера, тестировать кабельный сегмент и затем последовательно повторять эту процедуру для всех присоединенных узлов. Поэтому стоимость эксплуатации сети на тонком Ethernet обычно значительно превосходит стоимость эксплуатации аналогичной сети на витой паре, хотя капитальные затраты на кабельную систему для тонкого Ethernet обычно ниже.

Основные области применения оптоволоконного кабеля – вертикальная подсистема и подсистемы кампусов. Стоимость установки сетей на оптоволоконном кабеле для горизонтальной подсистемы оказывается весьма высокой. Эта стоимость складывается из стоимости сетевых адаптеров и стоимости монтажных работ, которая в случае оптоволокна гораздо выше, чем при работе с другими видами кабеля.

Преобладающим кабелем для горизонтальной подсистемы является неэкранированная витая пара категории 5. Ее позиции окончательно укрепились с принятием спецификации 802.3ab для применения на этом виде кабеля технологии Gigabit Ethernet.

Выбор типа кабеля для вертикальных подсистем

Кабель вертикальной (или магистральной) подсистемы, которая соединяет этажи здания, должен передавать данные на бóльшие расстояния и с большей скоростью по сравнению с кабелем горизонтальной подсистемы. Ранее основным видом кабеля для вертикальных подсистем был коаксиал. Теперь для этой цели все чаще используется оптоволоконный кабель.

Для вертикальной подсистемы выбор кабеля ограничивается двумя вариантами:

– оптоволокно – отличные характеристики пропускной способности, расстояния и защиты данных; устойчивость к электромагнитным помехам; может передавать голос, видеоизображение и данные. Но сравнительно дорого, сложно выполнять ответвления;

– толстый коаксиал – хорошие характеристики пропускной способности, расстояния и защиты данных. Но с ним сложно работать, хотя специалистов, имеющих подобный опыт работы, достаточно много.

Применение волоконно-оптического кабеля в вертикальной подсистеме имеет рад преимуществ. Он передает данные на значительно бóльшие расстояния без необходимости регенерации сигнала. Он имеет сердечник меньшего диаметра, поэтому может быть проложен в более узких местах. Так как передаваемые по нему сигналы являются световыми, а не электрическими, оптоволоконный кабель нечувствителен к электромагнитным и радиочастотным помехам, в отличие от медного коаксиального кабеля. Это делает оптоволоконный кабель идеальной средой передачи данных для промышленных сетей. Оптоволоконному кабелю не страшна молния, поэтому он хорош для внешней прокладки. Он обеспечивает более высокую степень защиты от несанкционированного доступа, т. к. ответвление гораздо легче обнаружить, чем в случае медного кабеля (при ответвлении резко уменьшается интенсивность света).

Оптоволоконный кабель имеет и недостатки. Он дороже, чем медный кабель, дороже обходится и его прокладка. Оптоволоконный кабель менее прочный, чем коаксиальный. Инструменты, применяемые при прокладке и тестировании оптоволоконного кабеля, имеют высокую стоимость и сложны в работе.

Толстый коаксиальный кабель также допустим в качестве магистрали сети, однако для новых кабельных систем более рационально использовать оптоволоконный кабель, т. к. он имеет больший срок службы и сможет в будущем поддерживать высокоскоростные и мультимедийные приложения. Причинами его повсеместного применения были широкая полоса пропускания, хорошая защищенность от электромагнитных помех и низкое радиоизлучение.

Хотя толстый коаксиальный кабель и дешевле, чем оптоволокно, но с ним гораздо сложнее работать. Он особенно чувствителен к различным уровням напряжения заземления, что часто бывает при переходе от одного этажа к другому. Эту проблему сложно разрешить. Поэтому основным кабелем для вертикальной подсистемы сегодня является волоконно-оптический кабель.

Выбор типа кабеля для подсистемы кампуса

Как и для вертикальных подсистем, оптоволоконный кабель является наилучшим выбором для подсистем нескольких зданий, расположенных в радиусе нескольких километров. Для этих подсистем также подходит толстый коаксиальный кабель. При выборе кабеля для кампуса нужно учитывать воздействие среды на кабель вне помещения. Для предотвращения поражения молнией лучше выбрать для внешней проводки неметаллический оптоволоконный кабель. По многим причинам внешний кабель производится в полиэтиленовой защитной оболочке высокой плотности. При подземной прокладке кабель должен иметь специальную влагозащитную оболочку (от дождя и подземной влаги), а также металлический защитный слой от грызунов и вандалов. Влагозащитный кабель имеет прослойку из инертного газа между диэлектриком, экраном и внешней оболочкой.

Кабель для внешней прокладки не подходит для прокладки внутри зданий, т. к. он выделяет при сгорании большое количество дыма.

2.19. Сетевые адаптеры и концентраторы

Концентраторы вместе с сетевыми адаптерами, а также кабельной системой представляют тот минимум оборудования, с помощью которого можно создать локальную сеть. Такая сеть будет представлять собой общую разделяемую среду. Концентраторы и сетевые адаптеры позволяют строить небольшие базовые фрагменты сетей, которые затем должны объединяться друг с другом с помощью мостов, коммутаторов и маршрутизаторов.

Функции сетевых адаптеров

Сетевой адаптер (Network Interface Card, NIC) вместе со своим драйвером реализует второй, канальный уровень модели открытых систем в конечном узле сети – компьютере. Более точно, в сетевой операционной системе пара адаптер и драйвер выполняет только функции физического и MAC-уровней, в то время как LLC-уровень обычно реализуется модулем операционной системы, единым для всех драйверов и сетевых адаптеров.

Сетевой адаптер совместно с драйвером выполняют две операции: передачу и прием кадра.

Передача кадра из компьютера в кабель состоит из перечисленных ниже этапов (некоторые могут отсутствовать, в зависимости от принятых методов кодирования):

– прием кадра данных LLC через межуровневый интерфейс вместе с адресной информацией MAC-уровня;

– оформление кадра данных MAC-уровня, в который инкапсулируется кадр LLC (с отброшенными флагами 01111110). Заполнение адресов назначения и источника, вычисление контрольной суммы;

– формирование символов кодов при использовании избыточных кодов типа 4B/5B. Скрэмблирование кодов для получения более равномерного спектра сигналов. Этот этап используется не во всех протоколах – например, технология Ethernet 10 Мбит/с обходится без него;

– выдача сигналов в кабель в соответствии с принятым линейным кодом – манчестерским, NRZI, MLT-3 и т. п.

Прием кадра из кабеля в компьютер включает следующие действия:

– прием из кабеля сигналов, кодирующих битовый поток;

– выделение сигналов на фоне шума. Эту операцию могут выполнять различные специализированные микросхемы или сигнальные процессоры DSP. В результате в приемнике адаптера образуется некоторая битовая последовательность, с большой степенью вероятности совпадающая с той, которая была послана передатчиком;

– если данные перед отправкой в кабель подвергались скрэмблированию, то они пропускаются через дескрэмблер, после чего в адаптере восстанавливаются символы кода, посланные передатчиком;

– проверка контрольной суммы кадра. Если она неверна, то кадр отбрасывается, а через межуровневый интерфейс наверх, протоколу LLC передается соответствующий код ошибки. Если контрольная сумма верна, то из MAC-кадра извлекается кадр LLC и передается через межуровневый интерфейс наверх, протоколу LLC.

Распределение обязанностей между сетевым адаптером и его драйвером стандартами не определяется, поэтому каждый производитель решает этот вопрос самостоятельно. Обычно сетевые адаптеры делятся на адаптеры для *клиентских компьютеров* и адаптеры для *серверов*.

В адаптерах для *клиентских компьютеров* значительная часть работы перекладывается на драйвер, тем самым адаптер оказывается проще и дешевле. Недостатком такого подхода является высокая степень загрузки центрального процессора компьютера рутинными работами по передаче кадров из оперативной памяти компьютера в сеть. Центральный процессор вынужден заниматься этой работой вместо выполнения прикладных задач пользователя.

Адаптеры, предназначенные для *серверов*, обычно снабжаются собственными процессорами, которые самостоятельно выполняют бóльшую часть работы по передаче кадров из оперативной памяти в сеть и в обратном направлении.

В зависимости от того, какой протокол реализует адаптер, адаптеры делятся на Ethernet-адаптеры, Token Ring-адаптеры, FDDI-адаптеры и т. д. Так как протокол Fast Ethernet позволяет за счет процедуры автопереговоров автоматически выбрать скорость работы сетевого адаптера в зависимости от возможностей концентратора, то многие адаптеры Ethernet поддерживают различные скорости работы.

Концентраторы. Основные и дополнительные функции концентраторов

Основная функция концентратора – это повторение кадра либо на всех портах (как определено в стандарте Ethernet), либо только на некоторых портах, в соответствии с алгоритмом, определенным соответствующим стандартом.

Концентратор обычно имеет несколько портов, к которым с помощью отдельных физических сегментов кабеля подключаются конечные узлы сети – компьютеры. Концентратор объединяет отдельные физические сегменты сети в единую разделяемую среду, доступ к которой осуществляется в соответствии с одним из протоколов локальных сетей – Ethernet, Token Ring и т. п. Так как логика доступа к разделяемой среде существенно зависит от технологии, то для каждого типа технологии выпускаются свои концентраторы – Ethernet; Token Ring; FDDI и 100VG-AnyLAN.

Каждый концентратор выполняет некоторую основную функцию, определенную в соответствующем протоколе той технологии, которую он поддерживает. Хотя эта функция достаточно детально определена в стандарте технологии, при ее реализации концентраторы разных производителей могут отличаться такими деталями, как количество портов, поддержка нескольких типов кабелей и т. п.

Кроме основной функции концентратор может выполнять некоторое количество дополнительных функций, которые либо в стандарте вообще не определены, либо являются факультативными. Например, концентратор Token Ring может выполнять функцию отключения некорректно работающих портов и перехода на резервное кольцо, хотя в стандарте такие его возможности не описаны.

Особенности реализации будут рассмотрены далее на примере концентраторов Ethernet.

В технологии Ethernet устройства, объединяющие несколько физических сегментов коаксиального кабеля в единую разделяемую среду, использовались давно и получили название «повторителей» по своей основной функции – повторение на всех своих портах сигналов, полученных на

входе одного из портов. В сетях на основе коаксиального кабеля обычными являлись двухпортовые повторители, соединяющие только два сегмента кабеля, поэтому термин *концентратор* к ним обычно не применялся.

С появлением спецификации 10Base-T для витой пары повторитель стал неотъемлемой частью сети Ethernet, т. к. без него связь можно было организовать только между двумя узлами сети. Многопортовые повторители Ethernet на витой паре стали называть концентраторами или хабами, т. к. в одном устройстве действительно концентрировались связи между большим количеством узлов сети. На рис. 2.53 показан типичный концентратор Ethernet, рассчитанный на образование небольших сегментов разделяемой среды. Он имеет 16 портов стандарта 10Base-T с разъемами RJ-45, а также один порт AUI для подключения внешнего трансивера. Обычно к этому порту подключается трансивер, работающий на коаксиал или оптоволокно. С помощью этого трансивера концентратор подключается к магистральному кабелю, соединяющему несколько концентраторов между собой, либо таким образом обеспечивается подключение станции, удаленной от концентратора более чем на 100 м.

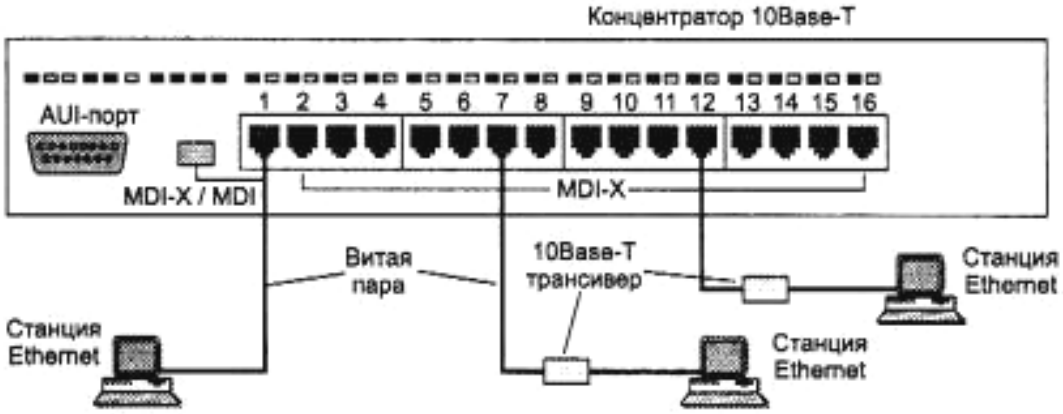


Рис. 2.53. Концентратор Ethernet

Для соединения концентраторов технологии 10Base-T между собой в иерархическую систему коаксиальный или оптоволоконный кабель необязателен, можно применять те же порты, что и для подключения конечных станций, с учетом одного обстоятельства. Дело в том, что обычный порт RJ-45, предназначенный для подключения сетевого адаптера и называемый MDI-X (кроссированный MDI), имеет инвертированную разводку контактов разъема, чтобы сетевой адаптер можно было подключить к концентратору с помощью стандартного соединительного кабеля, не кроссирующего контакты (рис. 2.54). В случае соединения концентраторов через стандарт-

ный порт MDI-X приходится использовать нестандартный кабель с перекрестным соединением пар. Поэтому некоторые изготовители снабжают концентратор выделенным портом MDI, в котором нет кроссирования пар. Таким образом, два концентратора можно соединить обычным некроссированным кабелем, если это делать через порт MDI-X одного концентратора и порт MDI второго. Чаще один порт концентратора может работать и как порт MDI-X, и как порт MDI, в зависимости от положения кнопочного переключателя, как это показано в нижней части рис. 2.54.

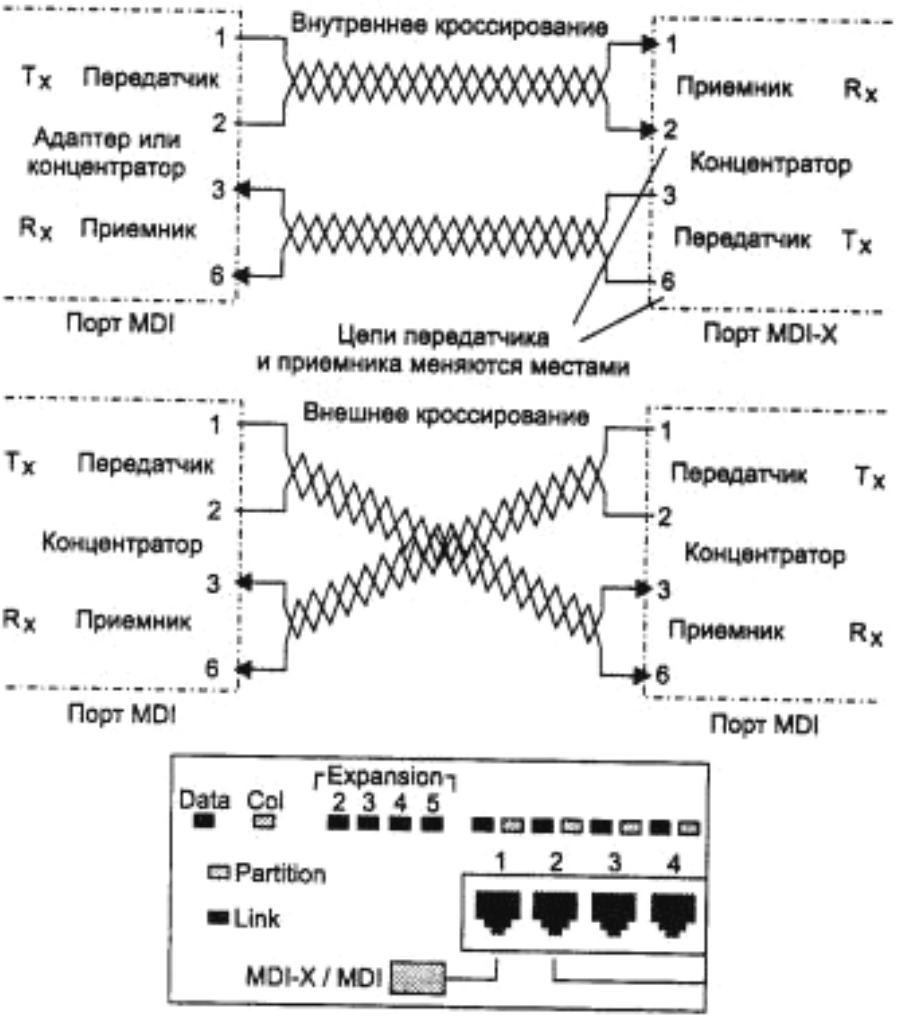


Рис. 2.54. Соединения типа «станция – концентратор» и «концентратор – концентратор» на витой паре

Многопортовый повторитель-концентратор Ethernet может по-разному рассматриваться при использовании правила 4-х хабов. В большинстве моделей все порты связаны с единственным блоком повторения и при прохождении сигнала между двумя портами повторителя блок повторения вносит задержку всего один раз. Поэтому такой концентратор нужно

считать одним повторителем с ограничениями, накладываемыми правилом 4-х хабов. Но существуют и другие модели повторителей, в которых на несколько портов имеется свой блок повторения. В таком случае каждый блок повторения нужно считать отдельным повторителем и учитывать его отдельно в правиле 4-х хабов.

Однако если существующие различия при выполнении основной функции концентраторов не столь велики, то их намного превосходит разброс в возможностях реализации концентраторами дополнительных функций.

Отключение портов

Очень полезной при эксплуатации сети является способность концентратора отключать некорректно работающие порты, изолируя тем самым остальную часть сети от возникших в узле проблем. Эту функцию называют автосегментацией (*autopartitioning*). Для концентратора FDDI эта функция для многих ошибочных ситуаций является основной, т. к. определена в протоколе. В то же время для концентратора Ethernet или Token Ring функция автосегментации для многих ситуаций является дополнительной, т. к. стандарт не описывает реакцию концентратора на эту ситуацию. Основной причиной отключения порта в стандартах Ethernet и Fast Ethernet является отсутствие ответа на последовательность импульсов link test, посылаемых во все порты каждые 16 мс. В этом случае неисправный порт переводится в состояние «отключен», но импульсы link test будут продолжать посылаться в порт с тем, чтобы при восстановлении устройства работа с ним была продолжена автоматически.

Ситуации, в которых концентраторы Ethernet и Fast Ethernet выполняют отключение порта:

– ошибки на уровне кадра. Если интенсивность прохождения через порт кадров, имеющих ошибки, превышает заданный порог, то порт отключается, а затем, при отсутствии ошибок в течение заданного времени, включается снова;

– множественные коллизии. Если концентратор фиксирует, что источником коллизии был один и тот же порт 60 раз подряд, то порт отключается. Через некоторое время порт снова будет включен;

– затянувшаяся передача (jabber). Как и сетевой адаптер, концентратор контролирует время прохождения одного кадра через порт. Если это время превышает время передачи кадра максимальной длины в 3 раза, то порт отключается.

Поддержка резервных связей

Так как использование резервных связей в концентраторах определено только в стандарте FDDI, то для остальных стандартов разработчики концентраторов поддерживают такую функцию с помощью своих частных решений. Например, концентраторы Ethernet/Fast Ethernet могут образовывать только иерархические связи без петель. Поэтому резервные связи всегда должны соединять отключенные порты, чтобы не нарушать логику работы сети. Обычно при конфигурировании концентратора администратор должен определить, какие порты являются основными, а какие по отношению к ним – резервными (рис. 2.55). Если по какой-либо причине порт отключается (срабатывает механизм автосегментации), концентратор делает активным его резервный порт.

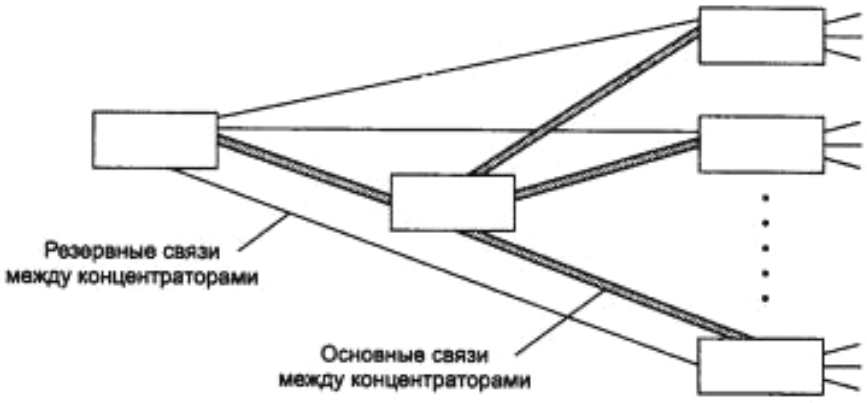


Рис. 2.55. Резервные связи между концентраторами Ethernet

Защита от несанкционированного доступа

Разделяемая среда предоставляет очень удобную возможность для несанкционированного прослушивания сети и получения доступа к передаваемым данным. Для этого достаточно подключить компьютер с программным анализатором протоколов к свободному разъему концентратора, записать на диск весь проходящий по сети трафик, а затем выделить из него нужную информацию.

Разработчики концентраторов предоставляют некоторый способ защиты данных в разделяемых средах.

Наиболее простой способ – назначение разрешенных MAC-адресов портам концентратора. В стандартном концентраторе Ethernet порты MAC-адресов не имеют. Защита заключается в том, что администратор вручную

связывает с каждым портом концентратора некоторый MAC-адрес. Этот MAC-адрес является адресом станции, которой разрешается подключаться к данному порту. Например, на рис. 2.56 первому порту концентратора назначен MAC-адрес 123 (условная запись). Компьютер с MAC-адресом 123 нормально работает с сетью через данный порт. Если злоумышленник отсоединяет этот компьютер и присоединяет вместо него свой, концентратор заметит, что при старте нового компьютера в сеть начали поступать кадры с адресом источника 789. Так как этот адрес является недопустимым для первого порта, то эти кадры фильтруются, порт отключается, а факт нарушения прав доступа может быть зафиксирован.

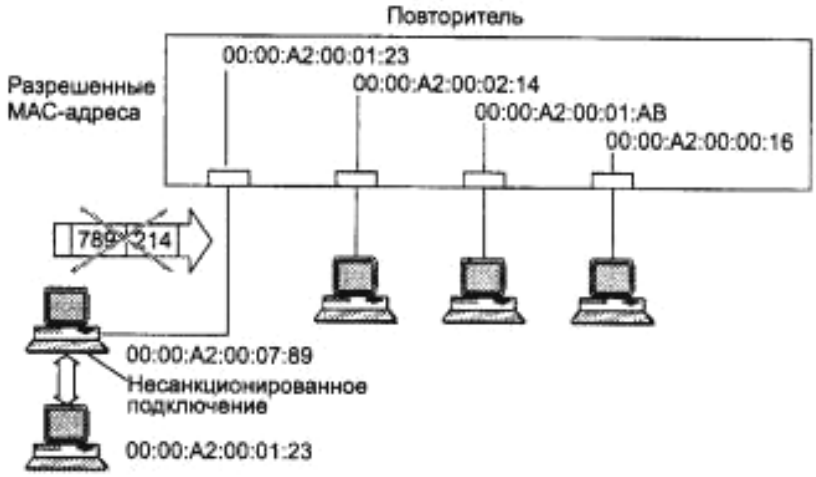


Рис. 2.56. Изоляция портов: передача кадров только от станций с фиксированными адресами

Заметим, что для реализации описанного метода защиты данных концентратор нужно предварительно сконфигурировать. Для этого концентратор должен иметь блок управления. Такие концентраторы обычно называют интеллектуальными. Блок управления представляет собой компактный вычислительный блок со встроенным программным обеспечением. Для взаимодействия администратора с блоком управления концентратор имеет консольный порт (чаще всего RS-232), к которому подключается терминал или персональный компьютер с программой эмуляции терминала. При присоединении терминала блок управления организует на его экране диалог, с помощью которого администратор вводит значения MAC-адресов. Блок управления может поддерживать и другие операции конфигурирования, например ручное отключение или включение портов и т. д. Для этого при подключении терминала блок управления выдает на экран некоторое меню, с помощью которого администратор выбирает нужное действие.

Другим способом защиты данных от несанкционированного доступа является их шифрация. Однако процесс истинной шифрации требует большой вычислительной мощности, и для повторителя, не буферизующего кадр, выполнить шифрацию «на лету» весьма сложно. Вместо этого в концентраторах применяется метод случайного искажения поля данных в пакетах, передаваемых портам с адресом, отличным от адреса назначения пакета. Этот метод сохраняет логику случайного доступа к среде, т. к. все станции видят занятость среды кадром информации, но только станция, которой послан этот кадр, может понять содержание поля данных кадра (рис. 2.57). Для реализации этого метода концентратор также нужно снабдить информацией о том, какие MAC-адреса имеют станции, подключенные к его портам. Обычно поле данных в кадрах, направляемых станциям, отличным от адресата, заполняется нулями.

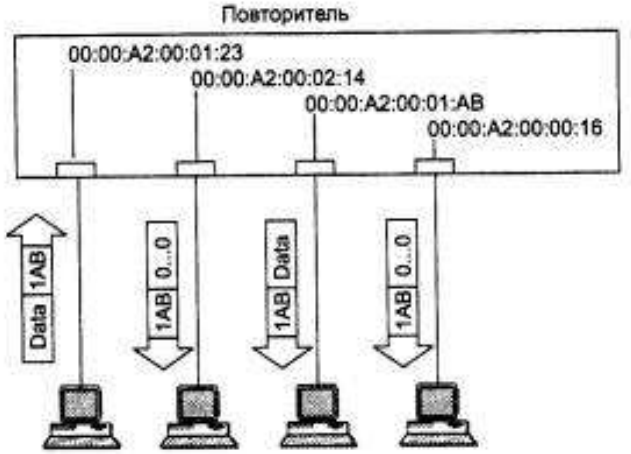


Рис. 2.57. Искажение поля данных в кадрах, не предназначенных для приема станциями

Многосегментные концентраторы

В многосегментых концентраторах имеются несколько несвязанных внутренних шин, которые предназначены для создания нескольких разделяемых сред. Например, концентратор, изображенный на рис. 2.58, имеет три внутренние шины Ethernet. Если, например, в таком концентраторе 72 порта, то каждый из этих портов может быть связан с любой из трех внутренних шин. На рисунке первые два компьютера связаны с шиной Ethernet 3, а третий и четвертый компьютеры – с шиной Ethernet 1. Первые два компьютера образуют один разделяемый сегмент, а третий и четвертый – другой разделяемый сегмент.

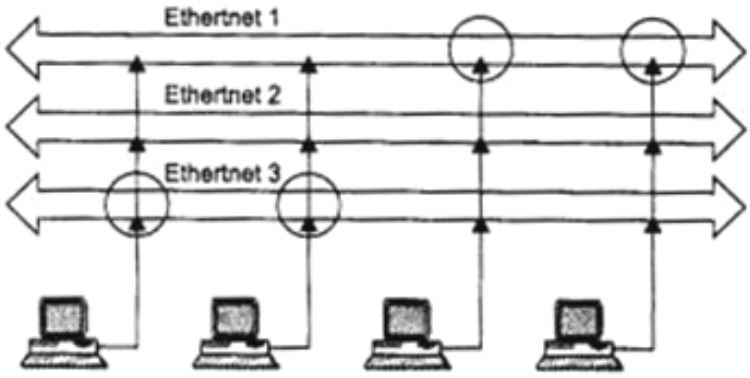


Рис. 2.58. Многосегментный концентратор

Между собой компьютеры, подключенные к разным сегментам, общаться через концентратор не могут, т. к. шины внутри концентратора никак не связаны.

Многосегментные концентраторы нужны для создания разделяемых сегментов, состав которых может легко изменяться. Большинство многосегментных концентраторов, например System 5000 компании Nortel Networks или PortSwitch Hub компании 3Com, позволяют выполнять операцию соединения порта с одной из внутренних шин чисто программным способом, например с помощью локального конфигурирования через консольный порт. В результате администратор сети может присоединять компьютеры пользователей к любым портам концентратора, а затем с помощью программы конфигурирования концентратора управлять составом каждого сегмента. Если завтра сегмент 1 станет перегруженным, то его компьютеры можно распределить между оставшимися сегментами концентратора.

Возможность многосегментного концентратора программно изменять связи портов с внутренними шинами называется конфигурационной коммутацией (*configuration switching*).

Многосегментные концентраторы – это программируемая основа больших сетей. Для соединения сегментов между собой нужны устройства другого типа – мосты/коммутаторы или маршрутизаторы. Такое межсетевое устройство должно подключаться к нескольким портам многосегментного концентратора, подсоединенным к разным внутренним шинам, и выполнять передачу кадров или пакетов между сегментами точно так же, как если бы они были образованы отдельными устройствами – концентраторами.

Для крупных сетей многосегментный концентратор играет роль интеллектуального кроссового шкафа, который выполняет новое соединение не за счет механического перемещения вилки кабеля в новый порт, а за счет программного изменения внутренней конфигурации устройства.

Конструктивное исполнение концентраторов

На конструктивное устройство концентраторов большое влияние оказывает их область применения. Концентраторы рабочих групп чаще всего выпускаются как устройства с фиксированным количеством портов, корпоративные концентраторы – как модульные устройства на основе шасси, а концентраторы отделов могут иметь стековую конструкцию. Такое деление не является жестким, и в качестве корпоративного концентратора может использоваться, например, модульный концентратор.

Концентратор с фиксированным количеством портов – это наиболее простое конструктивное исполнение, когда устройство представляет собой отдельный корпус со всеми необходимыми элементами (портами, органами индикации и управления, блоком питания), и эти элементы заменять нельзя. Обычно все порты такого концентратора поддерживают одну среду передачи, общее количество портов изменяется от 4 – 8 до 24. Один порт может быть специально выделен для подключения концентратора к магистрали сети или же для объединения концентраторов (в качестве такого порта часто используется порт с интерфейсом AUI, в этом случае применение соответствующего трансивера позволяет подключить концентратор к практически любой физической среде передачи данных).

Модульный концентратор выполняется в виде отдельных модулей с фиксированным количеством портов, устанавливаемых на общее шасси. Шасси имеет внутреннюю шину для объединения отдельных модулей в единый повторитель. Часто такие концентраторы являются многосегментными, тогда в пределах одного модульного концентратора работает несколько несвязанных между собой повторителей. Для модульного концентратора могут существовать различные типы модулей, отличающиеся количеством портов и типом поддерживаемой физической среды. Часто агент протокола SNMP выполняется в виде отдельного модуля, при установке которого концентратор превращается в интеллектуальное устройство. Модульные концентраторы позволяют более точно подобрать необходимую для конкретного применения конфигурацию концентратора, а также гибко и с минимальными затратами реагировать на изменения конфигурации сети.

Недостатком концентратора на основе шасси является высокая начальная стоимость такого устройства для случая, когда предприятию на первом этапе создания сети нужно установить всего 1 – 2 модуля. Высокая стоимость шасси вызвана тем, что оно поставляется вместе со всеми об-

щими устройствами, такими как избыточные источники питания и т. п. Поэтому в сетях средних размеров большую популярность завоевали стековые концентраторы.

Стековый концентратор, как и концентратор с фиксированным числом портов, выполнен в виде отдельного корпуса без возможности замены отдельных его модулей. Типичный вид нескольких стековых концентраторов Ethernet показан на рис. 2.59. Однако стековыми эти концентраторы называются не потому, что они устанавливаются один на другой. Такая чисто конструктивная деталь вряд ли удостоилась бы особого внимания, т. к. установка нескольких устройств одинаковых габаритных размеров в общую стойку практикуется очень давно. Стековые концентраторы имеют специальные порты и кабели для объединения нескольких таких корпусов в единый повторитель (рис. 2.60), который имеет общий блок повторения, обеспечивает общую ресинхронизацию сигналов для всех своих портов и поэтому с точки зрения правила 4-х хабов считается одним повторителем.

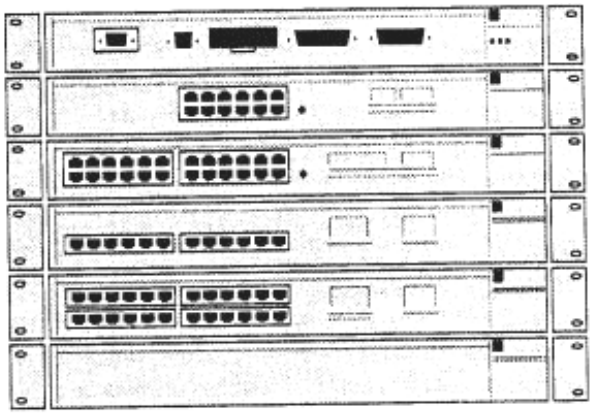


Рис. 2.59. Стековые концентраторы Ethernet

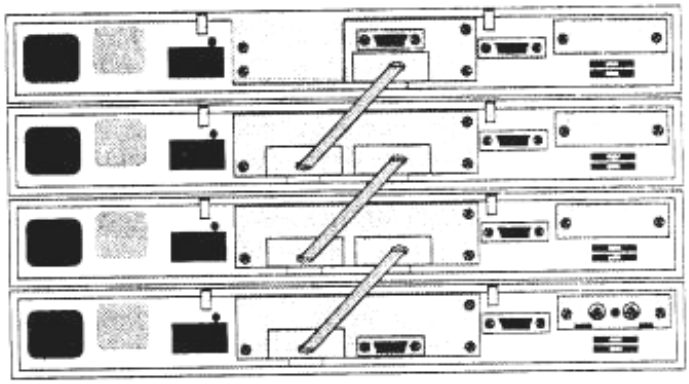


Рис. 2.60. Объединение стековых концентраторов в единое устройство с помощью специальных разъемов на задней панели

Если стековые концентраторы имеют несколько внутренних шин, то при соединении в стек эти шины объединяются и становятся общими для всех устройств стека. Число объединяемых в стек корпусов может быть достаточно большим (обычно до 8, но бывает и больше).

Модульно-стековые концентраторы представляют собой модульные концентраторы, объединенные специальными связями в стек. Как правило, корпуса таких концентраторов рассчитаны на небольшое количество модулей (1 – 3). Эти концентраторы сочетают достоинства концентраторов обоих типов.

2.20. Логическая структуризация сети с помощью мостов и коммутаторов

Под логической структуризацией сети понимается разбиение общей разделяемой среды на логические сегменты, которые представляют самостоятельные разделяемые среды с меньшим количеством узлов. Сеть, разделенная на логические сегменты, обладает более высокой производительностью и надежностью. Взаимодействие между логическими сегментами организуется с помощью мостов и коммутаторов.

Причины логической структуризации локальных сетей. Ограничения сети, построенной на общей разделяемой среде

При построении небольших сетей, состоящих из 10 – 30 узлов, использование стандартных технологий на разделяемых средах передачи данных приводит к экономичным и эффективным решениям.

Эффективность разделяемой среды для небольшой сети проявляется в первую очередь в следующих свойствах:

– простота топологии сети, допускающая легкое наращивание числа узлов (в небольших пределах);

– отсутствие потерь кадров из-за переполнения буферов коммуникационных устройств, т. к. новый кадр не передается в сеть, пока не принят предыдущий – сама логика разделения среды регулирует поток кадров и приостанавливает станции, слишком часто генерирующие кадры, заставляя их ждать доступа;

– простота протоколов, обеспечившая низкую стоимость сетевых адаптеров, повторителей и концентраторов.

Крупные сети, насчитывающие сотни и тысячи узлов, не могут быть построены на основе одной разделяемой среды даже такой скоростной

технологии, как Gigabit Ethernet. И не только потому, что практически все технологии ограничивают количество узлов в разделяемой среде: все виды семейства Ethernet – 1024 узлами, Token Ring – 260 узлами, а FDDI – 500 узлами. Даже сеть средних размеров, состоящая из 50 – 100 компьютеров и укладывающаяся в разрешенный максимум количества узлов, чаще всего будет плохо работать на одной разделяемой среде.

На рис. 2.61 показана зависимость задержек доступа к среде передачи данных в сетях Ethernet, Token Ring и FDDI от коэффициента использования сети *p*, который также часто называют коэффициентом нагрузки сети. Напомним, что коэффициент использования сети равен отношению трафика, который должна передать сеть, к ее максимальной пропускной способности. Для сети Ethernet максимальная пропускная способность равна 10 Мбит/с, а трафик, который она должна передать, равен сумме интенсивностей трафика, генерируемого каждым узлом сети. Коэффициент использования обычно измеряют в относительных единицах или процентах.

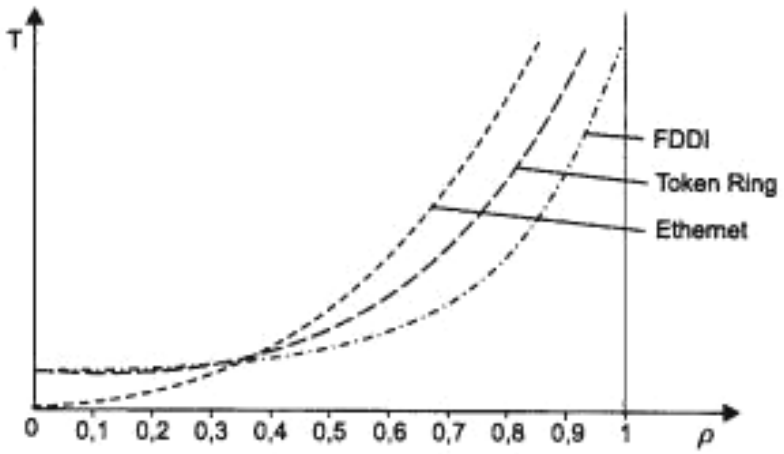


Рис. 2.61. Задержки доступа к среде передачи данных для технологий Ethernet, Token Ring и FDDI

Как видно из рисунка, всем технологиям присущ экспоненциальный рост величин задержек доступа при увеличении коэффициента использования сети, отличается только порог, при котором наступает резкий перелом в поведении сети, когда почти прямолинейная зависимость переходит в крутую экспоненту. Для всего семейства технологий Ethernet это 40 – 50%, для технологии Token Ring – 60 %, а технологии FDDI – 70%.

Влияние задержек и коллизий на полезную пропускную способность сети Ethernet отражает график, представленный на рис. 2.62.

Рис. 2.62. Зависимость полезной пропускной способности сети Ethernet от коэффициента использования

При загрузке сети до 50 % технология Ethernet на разделяемом сегменте хорошо справляется с передачей трафика, генерируемого конечными узлами. Однако при повышении интенсивности генерируемого узлами трафика сеть все больше времени начинает использовать неэффективно, повторно передавая кадры, которые вызвали коллизию. При возрастании интенсивности генерируемого трафика до такой величины, когда коэффициент использования сети приближается к 1, вероятность столкновения кадров настолько увеличивается, что практически любой кадр, который какая-либо станция пытается передать, сталкивается с другими кадрами, вызывая коллизию. Сеть перестает передавать полезную пользовательскую информацию и работает «на себя», обрабатывая коллизии.

Этот эффект известен на практике и исследован путем имитационного моделирования, поэтому сегменты Ethernet не рекомендуется загружать так, чтобы среднее значение коэффициента использования превосходило 30 %. Именно поэтому во многих системах управления сетями пороговая граница для индикатора коэффициента загрузки сети Ethernet по умолчанию устанавливается на величину 30 %.

Технология Ethernet наиболее чувствительна к перегрузкам разделяемого сегмента, но и другие технологии также страдают от этого эффекта, поэтому ограничения, связанные с возникающими коллизиями и большим временем ожидания доступа при значительной загрузке разделяемого сегмента, чаще всего оказываются более серьезными, чем ограничение на максимальное количество узлов, определенное в стандарте из соображений устойчивой передачи электрических сигналов в кабелях.

В результате даже сеть средних размеров трудно построить на одном разделяемом сегменте так, чтобы она работала эффективно при изменении интенсивности генерируемого станциями трафика. Кроме того, при использовании разделяемой среды накладываются жесткие ограничения на максимальный диаметр сети, который для всех технологий лежит в пределах нескольких километров (только технология FDDI позволяет строить локальные сети, длина которых измеряется десятками километров).

Преимущества логической структуризации сети

Ограничения, возникающие из-за использования общей разделяемой среды, можно преодолеть, разделив сеть на несколько разделяемых сред и соединив отдельные сегменты сети такими устройствами, как мосты, коммутаторы или маршрутизаторы (рис. 2.63).

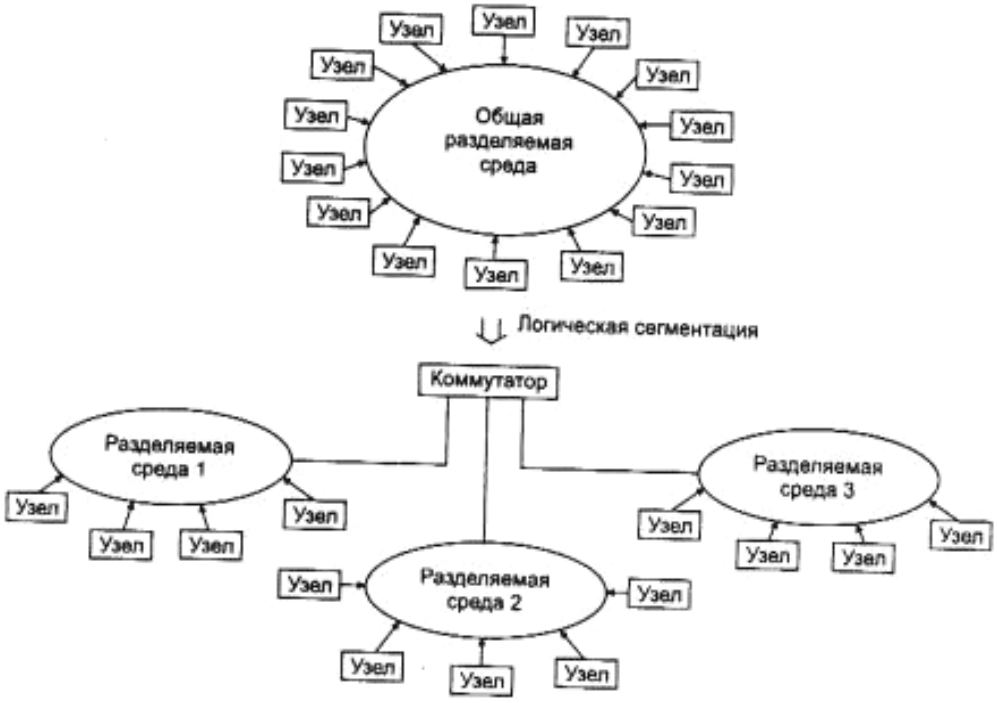


Рис. 2.63. Логическая структуризация сети

Перечисленные устройства передают кадры с одного своего порта на другой, анализируя адрес назначения, помещенный в этих кадрах. (В отличие от концентраторов, которые повторяют кадры на всех своих портах, передавая их во все подсоединенные к ним сегменты, независимо от того, в каком из них находится станция назначения.) Мосты и коммутаторы выполняют операцию передачи кадров на основе адресов канального уровня, т. е. MAC-адресов, а маршрутизаторы – на основе номера сети. При этом единая разделяемая среда, созданная концентраторами, делится на несколько

частей, каждая из которых присоединена к порту моста, коммутатора или маршрутизатора.

В этом случае сеть делится на логические сегменты или сеть подвергается *логической структуризации*. Логический сегмент представляет собой единую разделяемую среду. Деление сети на логические сегменты приводит к тому, что нагрузка, приходящаяся на каждый из вновь образованных сегментов, почти всегда оказывается меньше, чем нагрузка, которую испытывала исходная сеть. Следовательно, уменьшаются вредные эффекты от разделения среды: снижается время ожидания доступа, а в сетях Ethernet – и интенсивность коллизий.

Иллюстрация этого эффекта приведена на рис. 2.64. На нем изображены два сегмента, соединенные мостом. Внутри сегментов имеются повторители. До деления сети на сегменты весь трафик, генерируемый узлами сети, был общим и учитывался при определении коэффициента использования сети. Если обозначить среднюю интенсивность трафика, идущего от узла *i* к узлу *j* через C_{ij}, то суммарный трафик, который должна была передавать сеть до деления на сегменты, равен $C\sum = C_{ij}$ (суммирование проводится по всем узлам).

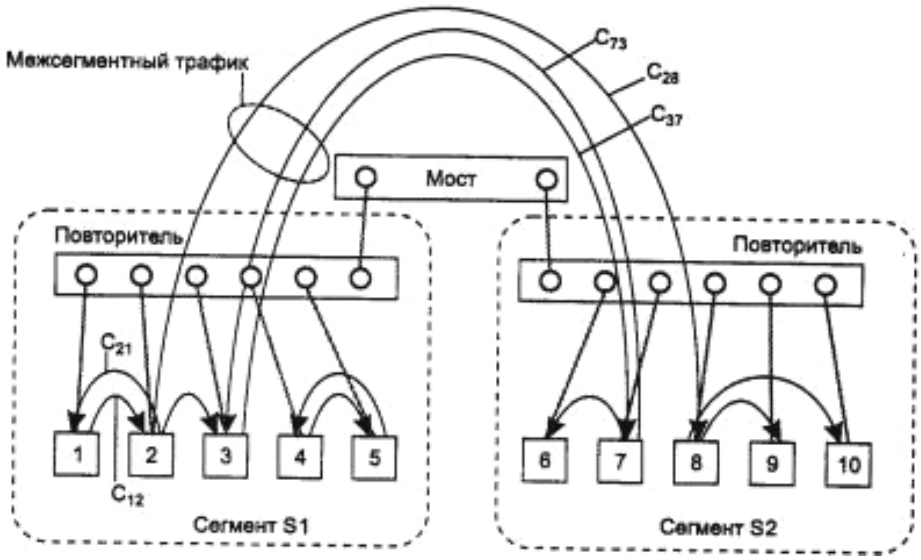


Рис. 2.64. Изменение нагрузки при делении сети на сегменты

После разделения сети на сегменты нагрузка каждого сегмента изменилась. При ее вычислении теперь нужно учитывать только внутрисегментный трафик, т. е. трафик кадров, которые циркулируют между узлами одного сегмента, а также межсегментный трафик, который либо направляется от узла данного сегмента узлу другого сегмента, либо приходит от уз-

ла другого сегмента в узел данного сегмента. Внутренний трафик другого сегмента теперь нагрузку на данный сегмент не создает.

Поэтому нагрузка, например, сегмента S1 стала равна $C_{S1} + C_{S1\text{-}S2}$, где C_{S1} – внутренний трафик сегмента S1, а $C_{S1\text{-}S2}$ – межсегментный трафик. Чтобы показать, что нагрузка сегмента S1 уменьшилась, заметим, что общую нагрузку сети до разделения на сегменты можно записать в такой форме: $C_{\sum} = C_{S1} + C_{S1\text{-}S2} + C_{S2}$, а значит, нагрузка сегмента S1 после разделения стала равной $C_{\sum} - C_{S2}$, т. е. уменьшилась на величину внутреннего трафика сегмента S2. А раз нагрузка на сегмент уменьшилась, то в соответствии с графиками, приведенными на рис. 2.61 и 2.62, задержки в сегментах также уменьшились, а полезная пропускная способность сегмента в целом и полезная пропускная способность, приходящаяся на один узел, увеличились.

На практике на предприятии всегда можно выделить группу компьютеров, которые принадлежат сотрудникам, выполняющим общую задачу. Это могут быть сотрудники одной рабочей группы, отдела, другого структурного подразделения предприятия. В большинстве случаев им нужен доступ к ресурсам сети их отдела и только изредка – доступ к удаленным ресурсам.

Большинство крупных сетей разрабатываются на основе структуры с общей магистралью, к которой через мосты и маршрутизаторы присоединяются подсети. Эти подсети обслуживают различные отделы. Подсети могут делиться и далее на сегменты, предназначенные для обслуживания рабочих групп.

В общем случае деление сети на логические сегменты повышает производительность сети (за счет разгрузки сегментов), а также гибкость построения сети, увеличивая степень защиты данных, и облегчает управление сетью.

Сегментация увеличивает гибкость сети. При построении сети как совокупности подсетей каждая подсеть может быть адаптирована к специфическим потребностям рабочей группы или отдела. Например, в одной подсети может использоваться технология Ethernet и ОС NetWare, а в другой – Token Ring и OS-400, в соответствии с традициями того или иного отдела или потребностями имеющихся приложений. Вместе с тем, у пользователей обеих подсетей есть возможность обмениваться данными через межсетевые устройства, такие как мосты, коммутаторы, маршрутизаторы. Процесс разбиения сети на логические сегменты можно рассматривать и в обратном направлении, как процесс создания большой сети из модулей уже имеющихся подсетей.

Подсети повышают безопасность данных. При подключении пользователей к различным физическим сегментам сети можно запретить доступ определенных пользователей к ресурсам других сегментов. Устанавливая различные логические фильтры на мостах, коммутаторах и маршрутизаторах, можно контролировать доступ к ресурсам, чего не позволяют сделать повторители.

Подсети упрощают управление сетью. Побочным эффектом уменьшения трафика и повышения безопасности данных является упрощение управления сетью. Проблемы очень часто локализуются внутри сегмента. Как и в случае структурированной кабельной системы, проблемы одной подсети не оказывают влияния на другие подсети. Подсети образуют логические домены управления сетью.

Сети должны проектироваться на двух уровнях: физическом и логическом. Логическое проектирование определяет места расположения ресурсов, приложений и способы группировки этих ресурсов в логические сегменты.

Структуризация с помощью мостов и коммутаторов

Здесь рассматриваются устройства логической структуризации сетей, работающие на канальном уровне стека протоколов, – мосты и коммутаторы. Структуризация сети возможна также на основе маршрутизаторов, которые для выполнения этой задачи привлекают протоколы сетевого уровня. Каждый способ структуризации – с помощью канального протокола и с помощью сетевого протокола – имеет свои преимущества и недостатки. В современных сетях часто используют комбинированный способ логической структуризации – небольшие сегменты объединяются устройствами канального уровня в более крупные подсети, которые, в свою очередь, соединяются маршрутизаторами.

Итак, сеть можно разделить на логические сегменты с помощью устройств двух типов – мостов (*bridge*) и/или коммутаторов (*switch, switching hub*). Сразу после появления коммутаторов в начале 90-х годов сложилось мнение, что мост и коммутатор – это принципиально различные устройства. И хотя постепенно представление о коммутаторах изменилось, это мнение можно услышать и сегодня.

Мост и коммутатор – это функциональные аналоги. Оба эти устройства продвигают кадры на основании одних и тех же алгоритмов. Мосты и коммутаторы используют два типа алгоритмов: алгоритм *прозрачного моста (transparent bridge)* либо алгоритм *моста с маршрутизацией от источника (source routing bridge)* для сетей Token Ring. Эти стандарты бы-

ли разработаны задолго до появления первого коммутатора, поэтому в них используется термин «мост». Когда же появилась первая промышленная модель коммутатора для технологии Ethernet, то она выполняла тот же алгоритм продвижения кадров, который был с десяток лет отработан мостами локальных и глобальных сетей. Точно так же поступают и все современные коммутаторы. Коммутаторы, которые продвигают кадры протокола Token Ring, работают по алгоритму Source Routing, характерному для мостов IBM.

Основное отличие коммутатора от моста заключается в том, что мост обрабатывает кадры последовательно, а коммутатор – параллельно. Это обстоятельство связано с тем, что мосты появились в те времена, когда сеть делили на небольшое количество сегментов, а межсегментный трафик был небольшим (он подчинялся правилу «80 на 20 %»). Сеть чаще всего делили на два сегмента, поэтому и термин был выбран соответствующий – мост. Для обработки потока данных со средней интенсивностью 1 Мбит/с мосту хватало производительности одного процессорного блока.

При изменении ситуации в конце 80-х – начале 90-х годов – появлении быстрых протоколов, производительных персональных компьютеров, мультимедийной информации, разделении сети на большое количество сегментов – классические мосты перестали справляться с работой. Обслуживание потоков кадров между теперь уже несколькими портами с помощью одного процессорного блока требовало значительного повышения быстродействия процессора, а это довольно дорогостоящее решение.

Более эффективным оказалось другое решение: для обслуживания потока, поступающего на каждый порт, в устройство ставился отдельный специализированный процессор, который реализовывал алгоритм моста. По сути, коммутатор – это мультипроцессорный мост, способный параллельно продвигать кадры сразу между всеми парами своих портов. Но если при добавлении процессорных блоков компьютер не перестали называть компьютером, а добавили только прилагательное «мультипроцессорный», то мультипроцессорные мосты стали называть «коммутаторами». Этому способствовал способ связи между отдельными процессорами коммутатора – они связывались коммутационной матрицей, похожей на матрицы мультипроцессорных компьютеров, связывающие процессоры с блоками памяти.

Постепенно коммутаторы вытеснили из локальных сетей классические однопроцессорные мосты. Основная причина этого – очень высокая производительность, с которой коммутаторы передают кадры между сегментами сети. Если мосты могли даже замедлять работу сети, когда их

производительность оказывалась меньше интенсивности межсегментного потока кадров, то коммутаторы всегда выпускаются с процессорами портов, которые могут передавать кадры с той максимальной скоростью, на которую рассчитан протокол. Добавление к этому параллельной передачи кадров между портами сделало производительность коммутаторов на несколько порядков выше, чем мостов, – коммутаторы могут передавать до нескольких миллионов кадров в секунду, в то время как мосты обычно обрабатывали 3 – 5 тысяч кадров в секунду.

Процесс вытеснения мостов начал протекать достаточно быстро с 1994 года, и в нынешнее время мосты практически не производятся сетевой индустрией. За время своего существования коммутаторы вобрали в себя многие дополнительные функции, которые появлялись в результате естественного развития сетевых технологий. К этим функциям относятся, например, поддержка виртуальных сетей (VLAN), приоритезация трафика, использование магистрального порта по умолчанию и т. п.

Сегодня мосты по-прежнему работают в сетях, но только на достаточно медленных глобальных связях между двумя удаленными локальными сетями. Такие мосты называются удаленными мостами (*remote bridge*).

Прозрачные мосты умеют, кроме передачи кадров в рамках одной технологии, транслировать протоколы локальных сетей, например, Ethernet в Token Ring, FDDI в Ethernet и т. п.

2.21. Принципы работы мостов

Алгоритм работы прозрачного моста

Прозрачные мосты незаметны для сетевых адаптеров конечных узлов, т. к. они самостоятельно строят специальную адресную таблицу, на основании которой можно решить, нужно передавать пришедший кадр в какой-либо другой сегмент или нет. Сетевые адаптеры при использовании прозрачных мостов работают точно так же, как и в случае их отсутствия, т. е. не предпринимают никаких дополнительных действий, чтобы кадр прошел через мост. Алгоритм прозрачного моста не зависит от технологии локальной сети, в которой устанавливается мост, поэтому прозрачные мосты Ethernet работают точно так же, как прозрачные мосты FDDI.

Прозрачный мост строит свою адресную таблицу на основании пассивного наблюдения за трафиком, циркулирующим в подключенных к его портам сегментах. При этом мост учитывает адреса источников кадров данных, поступающих на порты моста. По адресу источника кадра мост делает вывод о принадлежности этого узла тому или иному сегменту сети.

Рассмотрим процесс автоматического создания адресной таблицы моста и ее использования на примере простой сети, представленной на рис. 2.65.

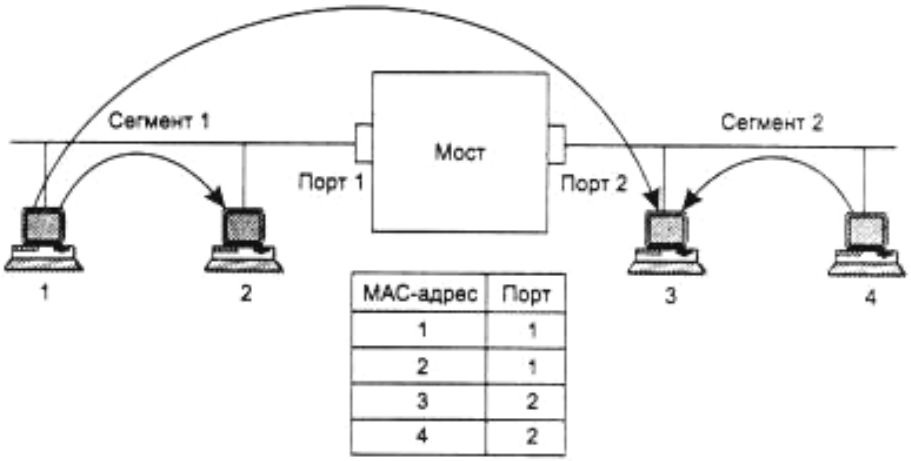


Рис. 2.65. Принцип работы прозрачного моста

Мост соединяет два логических сегмента. Сегмент 1 составляют компьютеры, подключенные с помощью одного отрезка коаксиального кабеля к порту 1 моста, а сегмент 2 – компьютеры, подключенные с помощью другого отрезка коаксиального кабеля к порту 2 моста.

Каждый порт моста работает как конечный узел своего сегмента за одним исключением – порт моста не имеет собственного MAC-адреса. Порт моста работает в так называемом *неразборчивом (promisquous)* режиме захвата пакетов, когда все поступающие на порт пакеты запоминаются в буферной памяти. С помощью такого режима мост следит за всем трафиком, передаваемым в присоединенных к нему сегментах, и использует проходящие через него пакеты для изучения состава сети. Так как в буфер записываются все пакеты, то адрес порта мосту не нужен.

В исходном состоянии мост ничего не знает о том, компьютеры с какими MAC-адресами подключены к каждому из его портов. Поэтому в этом случае мост просто передает любой захваченный и буферизованный кадр на все свои порты, за исключением того, от которого этот кадр получен. В нашем примере у моста только два порта, поэтому он передает кадры с порта 1 на порт 2, и наоборот. Отличие работы моста в этом режиме от повторителя в том, что он передает кадр не побитно, а с буферизацией. Буферизация разрывает логику работы всех сегментов как единой разделяемой среды. Когда мост собирается передать кадр с сегмента на сегмент, например с сегмента 1 на сегмент 2, он заново пытается получить доступ к сегменту 2 так же как конечный узел по правилам алгоритма доступа, в данном примере – по правилам алгоритма CSMA/CD.

Одновременно с передачей кадра на все порты мост изучает адрес источника кадра и делает новую запись о его принадлежности в своей адресной таблице, которую также называют таблицей фильтрации или маршрутизации. Например, получив на свой порт 1 кадр от компьютера 1, мост делает первую запись в своей адресной таблице: MAC-адрес 1 – порт 1. Если все четыре компьютера данной сети проявляют активность и посылают друг другу кадры, то скоро мост построит полную адресную таблицу сети, состоящую из 4-х записей – по одной записи на узел.

После того как мост прошел этап обучения, он может работать более рационально. При получении кадра, направленного, например, от компьютера 1 компьютеру 3, он просматривает адресную таблицу на предмет совпадения ее адресов с адресом назначения 3. Поскольку такая запись есть, то мост выполняет второй этап анализа таблицы – проверяет, находятся ли компьютеры с адресами источника (в нашем случае это адрес 1) и адресом назначения (адрес 3) в одном сегменте. Так как в нашем примере они находятся в разных сегментах, то мост выполняет операцию *продвижения (forwarding)* кадра – передает кадр на другой порт, предварительно получив доступ к другому сегменту.

Если бы оказалось, что компьютеры принадлежат одному сегменту, то кадр просто был бы удален из буфера и работа с ним на этом бы закончилась. Такая операция называется *фильтрацией (filtering)*.

Если же адрес назначения неизвестен, то мост передает кадр на все свои порты, кроме порта – источника кадра, как и на начальной стадии процесса обучения.

На самом деле процесс обучения никогда не заканчивается. Мост постоянно следит за адресами источника буферизуемых кадров, чтобы быть в состоянии автоматически приспосабливаться к изменениям, происходящим в сети, – перемещениям компьютеров из одного сегмента сети в другой, появлению новых компьютеров. С другой стороны, мост не ждет, когда адресная таблица заполнится полностью (да это и невозможно, поскольку заранее неизвестно, сколько компьютеров и адресов будут находиться в сегментах моста). Как только в таблице появляется первый адрес, мост пытается его использовать, проверяя совпадение с ним адресов назначения всех поступающих пакетов.

Входы адресной таблицы могут быть динамическими, создаваемыми в процессе самообучения моста, и статическими, создаваемыми вручную администратором сети. Динамические входы имеют срок жизни – при создании или обновлении записи в адресной таблице с ней связывается отметка

времени. По истечении определенного тайм-аута запись помечается как недействительная, если за это время мост не принял ни одного кадра с данным адресом в поле адреса источника. Это дает возможность автоматически реагировать на перемещения компьютера из сегмента в сегмент – при его отключении от старого сегмента запись о его принадлежности к нему со временем вычеркивается из адресной таблицы. После включения этого компьютера в работу в другом сегменте его кадры начнут попадать в буфер моста через другой порт и в адресной таблице появится новая запись, соответствующая текущему состоянию сети.

Статические записи не имеют срока жизни, что дает администратору возможность подправлять работу моста, если это необходимо.

Кадры с широковещательными MAC-адресами передаются мостом на все его порты, как и кадры с неизвестным адресом назначения. Такой режим распространения кадров называется *затоплением сети (flood)*. Наличие мостов в сети не препятствует распространению широковещательных кадров по всем сегментам сети, сохраняя ее прозрачность. Однако это является достоинством только в том случае, когда широковещательный адрес выработан корректно работающим узлом. Часто случается так, что в результате каких-либо программных или аппаратных сбоев протокол верхнего уровня или сам сетевой адаптер начинают работать некорректно и постоянно с высокой интенсивностью генерировать кадры с широковещательным адресом в течение длительного промежутка времени. Мост в этом случае передает эти кадры во все сегменты, затапливая сеть ошибочным трафиком. Такая ситуация называется *широковещательным штормом (broadcast storm)*.

К сожалению, мосты не защищают сети от широковещательного шторма, во всяком случае, по умолчанию, как это делают маршрутизаторы. Максимум, что может сделать администратор с помощью моста для борьбы с широковещательным штормом – установить для каждого узла предельно допустимую интенсивность генерации кадров с широковещательным адресом. Но при этом нужно точно знать, какая интенсивность является нормальной, а какая – ошибочной. При смене протоколов ситуация в сети может измениться, и то, что вчера считалось ошибочным, сегодня может оказаться нормой. Таким образом, мосты располагают весьма грубыми средствами борьбы с широковещательным штормом.

На рис. 2.66 показана типичная структура моста. Функции доступа к среде при приеме и передаче кадров выполняют микросхемы MAC, которые идентичны микросхемам сетевого адаптера.

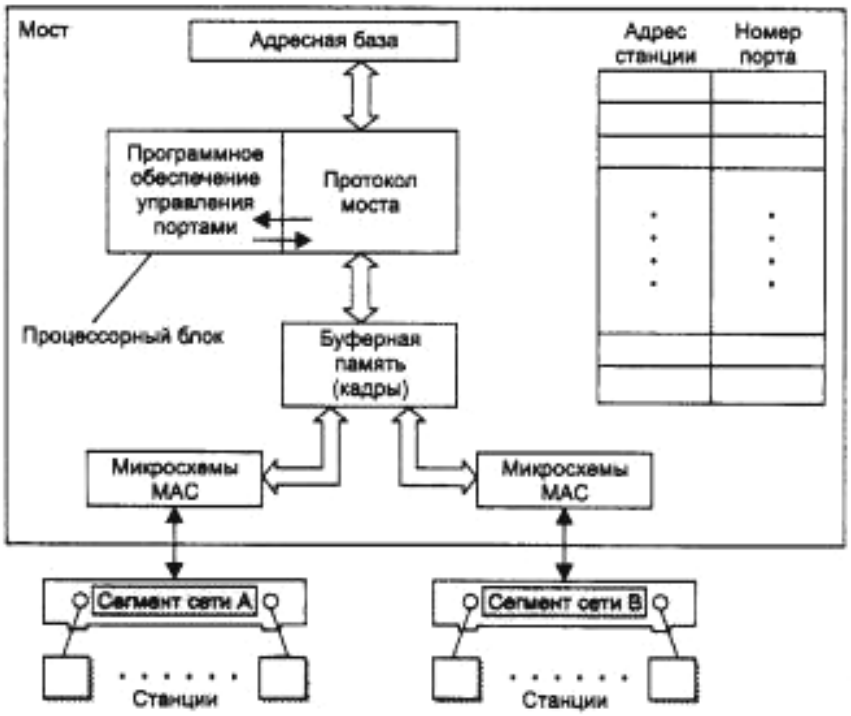


Рис. 2.66. Структура моста

Мосты с маршрутизацией от источника

Мосты с маршрутизацией от источника применяются для соединения колец Token Ring и FDDI, хотя для этих же целей могут использоваться и прозрачные мосты. *Маршрутизация от источника (Source Routing, SR)* основана на том, что станция-отправитель помещает в посылаемый в другое кольцо кадр всю адресную информацию о промежуточных мостах и кольцах, которые должен пройти кадр перед тем, как попасть в кольцо, к которому подключена станция-получатель. Хотя в название этого способа входит термин «маршрутизация», настоящей маршрутизации в строгом понимании этого термина здесь нет, т. к. мосты и станции по-прежнему используют для передачи кадров данных только информацию MAC-уровня, а заголовки сетевого уровня для мостов данного типа по-прежнему остаются неразличимой частью поля данных кадра.

Рассмотрим принципы работы мостов Source Routing (в дальнейшем – SR-мосты) на примере сети, изображенной на рис. 2.67. Сеть состоит из трех колец, соединенных тремя мостами. Для задания маршрута кольца и мосты имеют идентификаторы. SR-мосты не строят адресную таблицу, а при продвижении кадров пользуются информацией, имеющейся в соответствующих полях кадра данных.

Рис. 2.67. Мосты типа Source Routing

При получении каждого пакета SR-мосту нужно только просмотреть поле маршрутной информации (поле Routing Information Field, RIF, в кадре Token Ring или FDDI) на предмет наличия в нем своего идентификатора. И если он там присутствует и сопровождается идентификатором кольца, которое подключено к данному мосту, то в этом случае мост копирует поступивший кадр в указанное кольцо. В противном случае кадр в другое кольцо не копируется. В любом случае исходная копия кадра возвращается по исходному кольцу станции-отправителю, и если он был передан в другое кольцо, то бит A (адрес распознан) и бит C (кадр скопирован) поля статуса кадра устанавливаются в 1, чтобы сообщить станции-отправителю, что кадр был получен станцией назначения (в данном случае передан мостом в другое кольцо).

Так как маршрутная информация в кадре нужна не всегда, а только для передачи кадра между станциями, подключенными к разным кольцам, то наличие в кадре поля RIF обозначается установкой в 1 бит индивидуального/группового адреса (I/G) (при этом данный бит используется не по назначению, т. к. адрес источника всегда индивидуальный).

Для работы алгоритма маршрутизации от источника используются два дополнительных типа кадра: *одномаршрутный широковещательный кадр-исследователь SRBF (single-route broadcast frame)* и *многомаршрутный широковещательный кадр-исследователь ARBF (all-route broadcast frame)*.

Все SR-мосты должны быть сконфигурированы администратором вручную, чтобы передавать кадры ARBF на все порты, кроме порта – источника кадра, а для кадров SRBF некоторые порты мостов нужно заблокировать, чтобы в сети не было петель. В примере сети на рис. 2.67 для исключения петли администратор заблокировал оба порта моста 3 для передачи кадров SRBF.

Кадр первого типа отправляется станцией, когда она, во-первых, определяет, что станция назначения находится в другом кольце, а во-вторых, ей неизвестно, через какие мосты и кольца пролегает путь к этой станции назначения, т. е. неизвестен маршрут до этой станции. Первое обстоятельство выясняется, если кадр, отправленный по кольцу, возвращается в станцию-источник с неустановленными признаками распознавания адреса и копирования. Значит, ни одна из станций исходного кольца не является станцией назначения и кадр надо передавать по некоторому составному маршруту. Отсутствие маршрута к станции назначения в таблице моста является вторым обстоятельством, которое и вызывает отправку одномаршрутного кадра-исследователя SRBF.

В кадре SRBF станция задает длину поля RIF, равную нулю. Как и прозрачные мосты, SR-мосты работают в режиме «неразборчивого» захвата, буферизуя и анализируя все кадры. При получении кадра SRBF sr-мост передает его в исходном виде на все незаблокированные для этого типа кадров порты. Необходимость в конфигурировании топологии без петель для кадров-исследователей SRBF вызвана тем, что таким способом предотвращается возможность бесконечного зацикливания этих кадров.

В итоге кадр-исследователь SRBF, распространяясь по всем кольцам сети, доходит до станции назначения. В ответ станция назначения отправляет многомаршрутный широковещательный кадр-исследователь ARBF станции-отправителю. В отличие от кадра SRBF этот кадр передается мостами через все порты. При приеме такого кадра каждый промежуточный мост добавляет в поле маршрутной информации RIF новый описатель маршрута (свой идентификатор и идентификатор сегмента, с которого получен кадр), наращивает длину поля маршрутной информации и широковещательно его распространяет.

Для предотвращения зацикливания кадров ARBF мосты обрабатывают их следующим образом. Перед передачей кадра на какой-либо сегмент мост проверяет, нет ли идентификатора этого сегмента в списке маршрутов кадра. Если такой сегмент уже был пройден кадром, то кадр в данный сегмент не направляется.

Станция-источник получает в общем случае несколько кадров-ответов, прошедших по всем возможным маршрутам составной сети, и выбирает наилучший маршрут (обычно по количеству пересечений промежуточных мостов). Именно для получения информации о всех возможных маршрутах кадр ARBF передается по всем возможным направлениям.

Затем маршрутная информация помещается в таблицу маршрутизации станции и используется для отправки кадров данных станции назначения по наилучшему маршруту за счет помещения последовательности номеров сетей и мостов в заголовке каждого такого кадра.

Ограничения топологии сети, построенной на мостах

Слабая защита от широковещательного шторма – одно из главных ограничений моста, но не единственное. Другим серьезным ограничением их функциональных возможностей является невозможность поддержки петлеобразной конфигурации сети. Рассмотрим это ограничение на примере сети, изображенной на рис. 2.68.

Два сегмента параллельно соединены двумя мостами, так что образовалась активная петля. Пусть новая станция с адресом 10 впервые начинает работу в данной сети. Обычно начало работы любой операционной системы сопровождается рассылкой широковещательных кадров, в которых станция заявляет о своем существовании и одновременно ищет серверы сети.

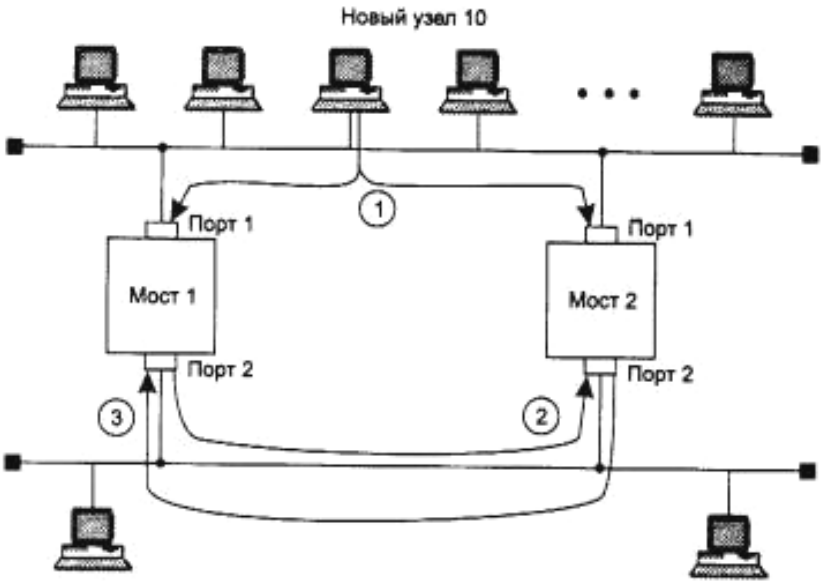


Рис. 2.68. Влияние замкнутых маршрутов на работу мостов

На этапе 1 станция посылает первый кадр с широковещательным адресом назначения и адресом источника 10 в свой сегмент. Кадр попадает как в мост 1, так и в мост 2. В обоих мостах новый адрес источника 10 заносится в адресную таблицу с пометкой о его принадлежности сегменту 1.

Так как адрес назначения широковещательный, то каждый мост должен передать кадр на сегмент 2. Эта передача происходит поочередно, в соответствии с методом случайного доступа технологии Ethernet. Пусть первым доступ к сегменту 2 получил мост 1 (этап 2 на рис. 2.68). При появлении пакета на сегменте 2 мост 2 принимает его в свой буфер и обрабатывает. Он видит, что адрес 10 уже есть в его адресной таблице, но пришедший кадр является более свежим, и он утверждает, что адрес 10 принадлежит сегменту 2, а не 1. Поэтому мост 2 корректирует содержимое базы и делает запись о том, что адрес 10 принадлежит сегменту 2.Результаты наличия петли:

– «размножение» кадра, т. е. появление нескольких его копий (в данном случае – двух, но если бы сегменты были соединены тремя мостами, то трех и т. д.);

– бесконечная циркуляция обеих копий кадра по петле в противоположных направлениях, а значит, засорение сети ненужным трафиком;

– постоянная перестройка мостами своих адресных таблиц, т. к. кадр с адресом источника 10 будет появляться то на одном порту, то на другом.

Чтобы исключить все эти нежелательные эффекты, мосты нужно применять так, чтобы между логическими сегментами не было петель, т. е. строить с помощью мостов только древовидные структуры, гарантирующие наличие только одного пути между любыми двумя сегментами. Тогда кадры от каждой станции будут поступать в мост всегда с одного и того же порта и мост сможет правильно решать задачу выбора рационального маршрута в сети.

Ограничение топологии структурированной сети древовидной структурой вытекает из самого принципа построения адресной таблицы мостом, а поэтому точно так же это ограничение действует и на коммутаторы.

В простых сетях сравнительно легко гарантировать существование одного и только одного пути между двумя сегментами. Но когда количество соединений возрастает и сеть становится сложной, то вероятность непреднамеренного образования петли оказывается высокой. Кроме того, желательно для повышения надежности иметь между мостами резервные связи, которые не участвуют при нормальной работе основных связей в передаче информационных пакетов станций, но при отказе какой-либо основной связи образуют новую связную рабочую конфигурацию без петель.

Поэтому в сложных сетях между логическими сегментами прокладывают избыточные связи, которые образуют петли, но для исключения активных петель блокируют некоторые порты мостов. Наиболее просто эта задача решается вручную, но существуют и алгоритмы, которые позволя-

ют решать ее автоматически. Наиболее известным является стандартный алгоритм покрывающего дерева (Spanning Tree Algorithm, STA). Кроме того, имеются фирменные алгоритмы, решающие ту же задачу, но с некоторыми улучшениями для конкретных моделей коммутаторов.

2.22. Коммутаторы локальных сетей

Технология коммутации сегментов Ethernet была предложена фирмой Kalpana в 1990 году в ответ на растущие потребности в повышении пропускной способности связей высокопроизводительных серверов с сегментами рабочих станций.

Структурная схема коммутатора EtherSwitch, предложенного фирмой Kalpana, представлена на рис. 2.69.

Каждый из 8 портов 10Base-T обслуживается одним процессором пакетов Ethernet – EPP (Ethernet Packet Processor). Кроме того, коммутатор имеет системный модуль, который координирует работу всех процессоров EPP. Системный модуль ведет общую адресную таблицу коммутатора и обеспечивает управление коммутатором по протоколу SNMP. Для передачи кадров между портами используется коммутационная матрица.

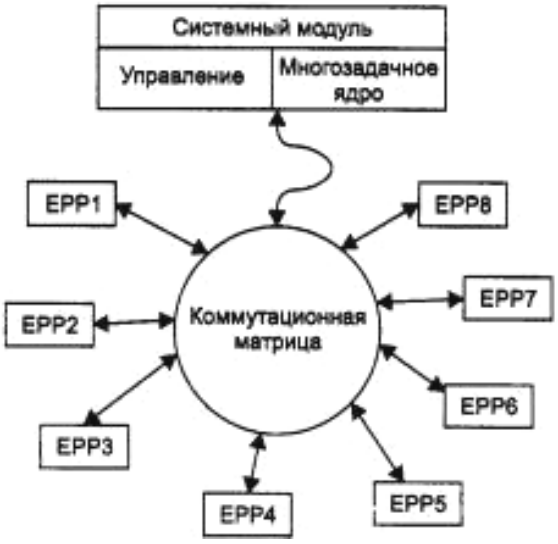


Рис. 2.69. Структура коммутатора EtherSwitch компании Ka1pana

Коммутационная матрица работает по принципу коммутации каналов. Для 8-и портов матрица может обеспечить 8 одновременных внутренних каналов при полудуплексном режиме работы портов и 16 – при полнодуплексном, когда передатчик и приемник каждого порта работают независимо друг от друга.

При поступлении кадра в какой-либо порт процессор EPP буферизует несколько первых байтов кадра, чтобы прочитать адрес назначения. После получения адреса назначения процессор сразу же принимает решение о передаче пакета, не дожидаясь прихода остальных байтов кадра. Для этого он просматривает свой собственный кэш адресной таблицы, а если не находит там нужного адреса, обращается к системному модулю, который работает в многозадачном режиме, параллельно обслуживая запросы всех процессоров EPP. Системный модуль производит просмотр общей адресной таблицы и возвращает процессору найденную строку, которую тот буферизует в своем кэше для последующего использования.

После нахождения адреса назначения процессор EPP знает, что нужно дальше делать с поступающим кадром (во время просмотра адресной таблицы процессор продолжал буферизацию поступающих в порт байтов кадра). Если кадр нужно отфильтровать, процессор просто прекращает записывать в буфер байты кадра, очищает буфер и ждет поступления нового кадра.

Если же кадр нужно передать на другой порт, то процессор обращается к коммутационной матрице и пытается установить в ней путь, связывающий его порт с портом, через который идет маршрут к адресу назначения. Коммутационная матрица может это сделать только в том случае, когда порт адреса назначения в этот момент свободен, т. е. не соединен с другим портом.

Если же порт занят, то, как и в любом устройстве с коммутацией каналов, матрица в соединении отказывает. В этом случае кадр полностью буферизуется процессором входного порта, после чего процессор ожидает освобождения выходного порта и образования коммутационной матрицей нужного пути.

После того как нужный путь установлен, в него направляются буферизованные байты кадра, которые принимаются процессором выходного порта. Как только процессор выходного порта получает доступ к подключенному к нему сегменту Ethernet по алгоритму CSMA/CD, байты кадра сразу же начинают передаваться в сеть. Процессор входного порта постоянно хранит несколько байт принимаемого кадра в своем буфере, что позволяет ему независимо и асинхронно принимать и передавать байты кадра (рис. 2.70).

При свободном в момент приема кадра состоянии выходного порта задержка между приемом первого байта кадра коммутатором и появлением

этого же байта на выходе порта адреса назначения составляла у коммутатора компании Kalpana всего 40 мкс, что было гораздо меньше задержки кадра при его передаче мостом.

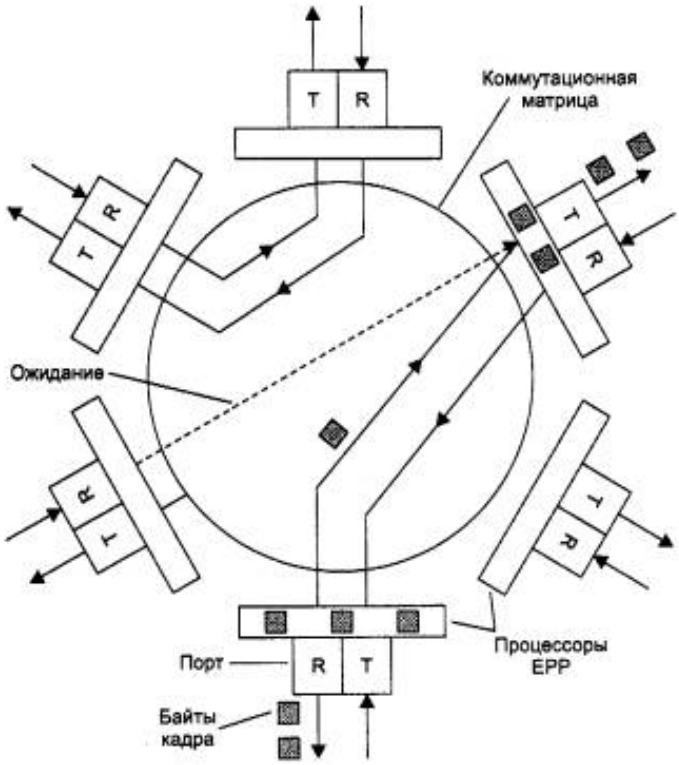


Рис. 2.70. Передача кадра через коммутационную матрицу

Описанный способ передачи кадра без его полной буферизации получил название коммутации «на лету» («on-the-fly») или «напролет» («cut-through»). Этот способ представляет, по сути, конвейерную обработку кадра, когда частично совмещаются во времени несколько этапов его передачи (рис. 2.71):

1) прием первых байтов кадра процессором входного порта, включая прием байтов адреса назначения;

2) поиск адреса назначения в адресной таблице коммутатора (в кэше процессора или в общей таблице системного модуля);

3) коммутация матрицы;

4) прием остальных байтов кадра процессором входного порта;

5) прием байтов кадра (включая первые) процессором выходного порта через коммутационную матрицу;

6) получение доступа к среде процессором выходного порта;

7) передача байтов кадра процессором выходного порта в сеть.

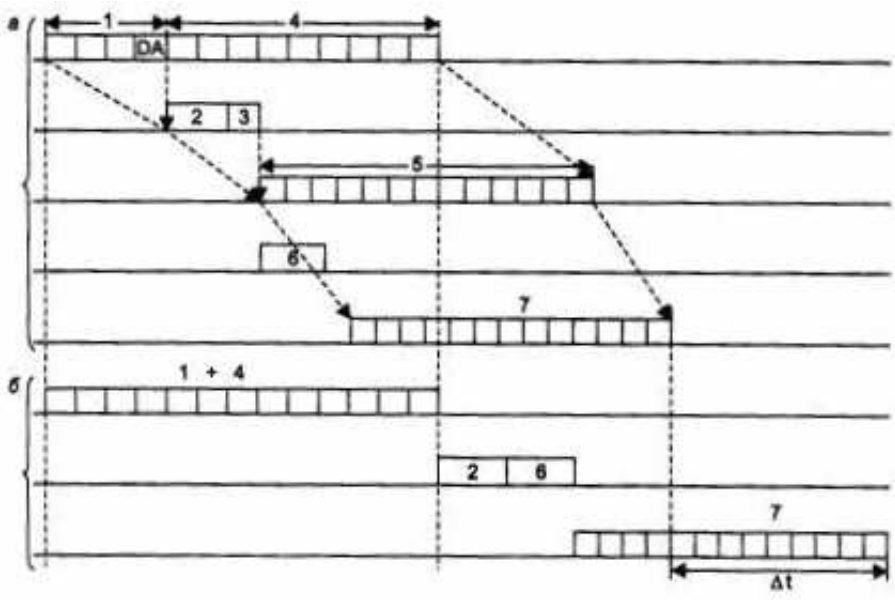


Рис. 2.71. Экономия времени при конвейерной обработке кадра:
а – конвейерная обработка; *б* – обычная обработка с полной буферизацией

Этапы 2 и 3 совместить во времени нельзя, т. к. без знания номера выходного порта операция коммутации матрицы не имеет смысла.

По сравнению с режимом полной буферизации кадра, также приведенным на рис. 2.71, экономия от конвейеризации получается ощутимой.

Однако главной причиной повышения производительности сети при использовании коммутатора является *параллельная обработка* нескольких кадров.

Этот эффект иллюстрирует рис. 2.72. На рисунке изображена идеальная в отношении повышения производительности ситуация, когда четыре порта из восьми передают данные с максимальной для протокола Ethernet скоростью 10 Мб/с, причем они передают эти данные на остальные четыре порта коммутатора не конфликтуя – потоки данных между узлами сети распределились так, что для каждого принимающего кадры порта есть свой выходной порт. Если коммутатор успевает обрабатывать входной трафик даже при максимальной интенсивности поступления кадров на входные порты, то общая производительность коммутатора в приведенном примере составит $4 \cdot 10 = 40$ Мбит/с, а при обобщении примера для N портов $(N/2) \cdot 10$ Мбит/с. Говорят, что коммутатор предоставляет каждой станции или сегменту, подключенным к его портам, выделенную пропускную способность протокола.

Естественно, что в сети не всегда складывается такая ситуация, которая изображена на рис. 2.72. Если двум станциям, например станциям, подключенным к портам 3 и 4, одновременно нужно записывать данные на один и тот же сервер, подключенный к порту 8, то коммутатор не сможет выделить каждой станции поток данных по 10 Мбит/с, т. к. порт 8 не может передавать данные со скоростью 20 Мбит/с. Кадры станций будут ожидать во внутренних очередях входных портов 3 и 4, когда освободится порт 8 для передачи очередного кадра. Очевидно, хорошим решением для такого распределения потоков данных было бы подключение сервера к более высокоскоростному порту, например Fast Ethernet.

Широкому применению коммутаторов способствовало то обстоятельство, что внедрение технологии коммутации не требовало замены установленного в сетях оборудования – сетевых адаптеров, концентраторов, кабельной системы. Порты коммутаторов работали в обычном полудуплексном режиме, поэтому к ним прозрачно можно было подключить как конечный узел, так и концентратор, организующий целый логический сегмент.

Так как коммутаторы и мосты прозрачны для протоколов сетевого уровня, то их появление в сети не оказало никакого влияния на маршрутизаторы сети, если они там имелись.

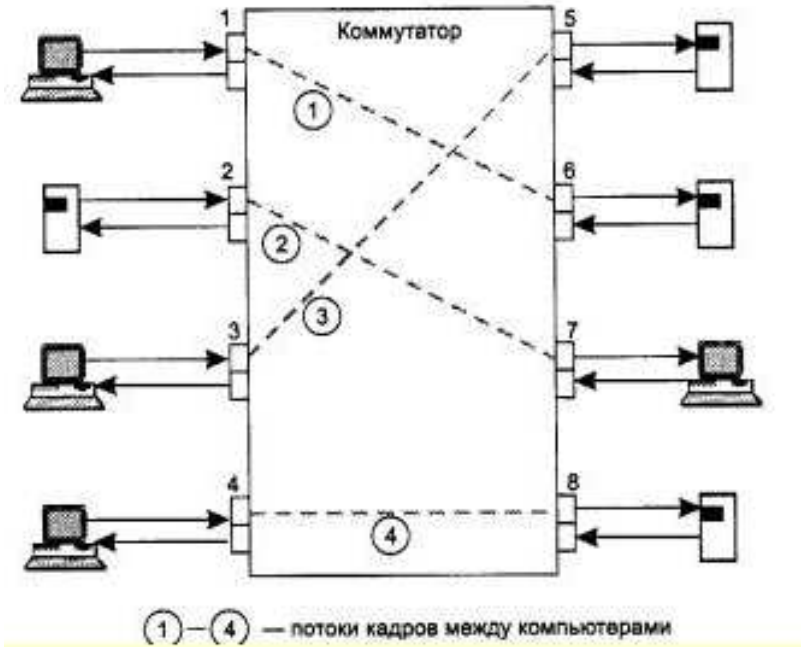


Рис. 2.72. Параллельная передача кадров коммутатором

Удобство использования коммутатора состоит еще и в том, что это самообучающееся устройство и если администратор не нагружает его дополнительными функциями, конфигурировать его не обязательно – нужно только правильно подключить разъемы кабелей к портам коммутатора, а дальше он будет работать самостоятельно и эффективно выполнять поставленную перед ним задачу повышения производительности сети.

Изменения в работе MAC-уровня при полнодуплексной работе

Технология коммутации сама по себе не имеет непосредственного отношения к методу доступа к среде, который используется портами коммутатора. При подключении сегментов, представляющих собой разделяемую среду, порт коммутатора должен поддерживать полудуплексный режим, т. к. является одним из узлов этого сегмента.

Однако когда к каждому порту коммутатора подключен не сегмент, а только один компьютер, причем по двум раздельным каналам, как это происходит почти во всех стандартах физического уровня, кроме коаксиальных версий Ethernet, ситуация становится не такой однозначной. Порт может работать как в обычном полудуплексном режиме, так и в полнодуплексном. Подключение к портам коммутатора не сегментов, а отдельных компьютеров называется *микросегментацией*.

В обычном режиме работы порт коммутатора по-прежнему распознает коллизии. Доменом коллизий в этом случае будет участок сети, включающий передатчик коммутатора, приемник коммутатора, передатчик сетевого адаптера компьютера, приемник сетевого адаптера компьютера и две витые пары, соединяющие передатчики с приемниками (рис. 2.73).

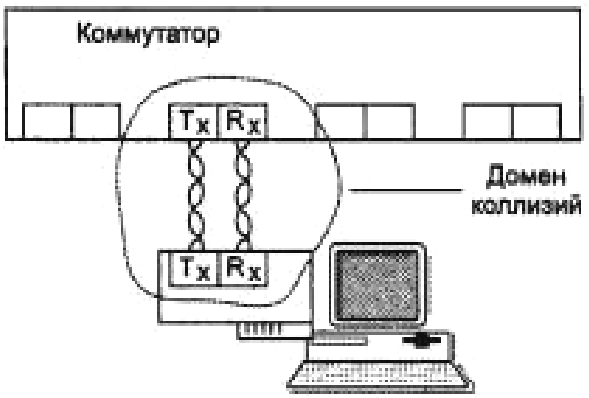


Рис. 2.73. Домен коллизий, образуемый компьютером и портом коммутатора

Коллизия возникает, когда передатчики порта коммутатора и сетевого адаптера одновременно или почти одновременно начинают передачу своих кадров, считая, что изображенный на рисунке сегмент свободен. Правда, вероятность коллизии в таком сегменте гораздо меньше, чем в сегменте, состоящем из 20 – 30 узлов, но она не нулевая.

В полнодуплексном режиме одновременная передача данных передатчиком порта коммутатора и сетевого адаптера коллизией не считается. В принципе, это достаточно естественный режим работы для индивидуальных полнодуплексных каналов связи, и он часто используется в протоколах территориальных сетей. При полнодуплексной связи порты Ethernet могут передавать данные со скоростью 20 Мбит/с – по 10 Мбит/с в каждом направлении.

Естественно, необходимо, чтобы MAC-узлы взаимодействующих устройств поддерживали этот специальный режим. В случае когда только один узел будет поддерживать полнодуплексный режим, второй узел будет постоянно фиксировать коллизии и приостанавливать свою работу, в то время как другой узел будет продолжать передавать данные, которые никто в этот момент не принимает. Изменения, которые нужно сделать в логике MAC-узла, чтобы он мог работать в полнодуплексном режиме, минимальны – нужно просто отменить фиксацию и отработку коллизий в сетях Ethernet, а в сетях Token Ring и FDDI посылать кадры в коммутатор, не дожидаясь прихода токена доступа, а тогда, когда это нужно конечному узлу. Фактически, при работе в полнодуплексном режиме MAC-узел не использует метод доступа к среде, разработанный для данной технологии.

Так как переход на полнодуплексный режим работы требует изменения логики работы MAC-узлов и драйверов сетевых адаптеров, то он сначала был опробован при соединении двух коммутаторов. Уже первые модели коммутатора EtherSwitch компании Kalpana поддерживали полнодуплексный режим при взаимном соединении, обеспечивая скорость взаимного обмена 20 Мбит/с.

После опробования полнодуплексной технологии на соединениях коммутатор – коммутатор разработчики реализовали ее и в сетевых адаптерах, в основном, в адаптерах Ethernet и Fast Ethernet. При разработке технологий Fast Ethernet и Gigabit Ethernet полнодуплексный режим стал одним из двух полноправных стандартных режимов работы узлов сети. Многие сетевые адаптеры сейчас могут поддерживать оба режима работы, отрабатывая логику алгоритма доступа CSMA/CD при подключении к порту концентратора и работая в полнодуплексном режиме при подключении к порту коммутатора.

Проблема управления потоком данных при полнодуплексной работе

Простой отказ от поддержки алгоритма доступа к разделяемой среде без какой-либо модификации протокола ведет к повышению вероятности потерь кадров коммутаторами, т. к. при этом теряется контроль за потоками кадров, направляемых конечными узлами в сеть. Раньше поток кадров регулировался методом доступа к разделяемой среде, так что слишком часто генерирующий кадры узел вынужден был ждать своей очереди к среде и фактическая интенсивность потока данных, который направлял в сеть этот узел, была заметно меньше той интенсивности, которую узел хотел бы отправить в сеть. При переходе на полнодуплексный режим узлу разрешается отправлять кадры в коммутатор всегда, когда это ему нужно, поэтому коммутаторы сети могут в этом режиме сталкиваться с перегрузками, не имея при этом никаких средств регулирования потока кадров.

Поэтому если входной трафик неравномерно распределяется между выходными портами, легко представить ситуацию, когда в какой-либо выходной порт коммутатора будет направляться трафик с суммарной средней интенсивностью большей, чем протокольный максимум. На рис. 2.74 изображена как раз такая ситуация, когда в порт 3 коммутатора направляется трафик от портов 1, 2, 4 и 6 с суммарной интенсивностью в 22100 кадров в секунду. Порт 3 оказывается загружен на 150 %. Естественно, что когда кадры поступают в буфер порта со скоростью 20100 кадров в секунду, а уходят со скоростью 14880 кадров в секунду, то внутренний буфер выходного порта начинает неуклонно заполняться необработанными кадрами.

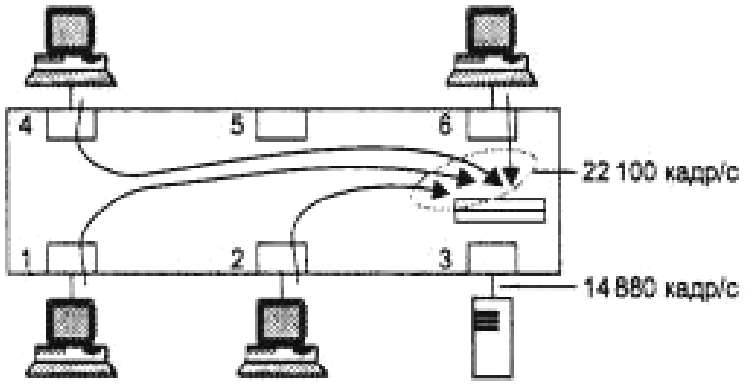


Рис. 2.74. Переполнение буфера порта из-за несбалансированности трафика

Какой бы ни был объем буфера порта, он в какой-то момент времени обязательно переполнится. Нетрудно подсчитать, что при размере буфера в 100 Кбайт в приведенном примере полное заполнение буфера произойдет

через 0,22 секунды после начала его работы (буфер такого размера может хранить до 1600 кадров размером в 64 байт). Увеличение буфера до 1 Мбайт даст увеличение времени заполнения буфера до 2,2 секунд, что также неприемлемо. А потери кадров всегда очень нежелательны, т. к. это снижает полезную производительность сети и коммутатор, теряющий кадры, может значительно ухудшить производительность сети вместо ее улучшения.

В глобальных сетях коммутаторы технологии X.25 поддерживают протокол канального уровня LAP-B, который имеет специальные кадры управления потоком «Приемник готов» (RR) и «Приемник не готов» (RNR). Протокол LAP-B работает между соседними коммутаторами сети X.25 и в том случае, когда очередь коммутатора доходит до опасной границы, запрещает своим ближайшим соседям с помощью кадра «Приемник не готов» передавать ему кадры, пока очередь не уменьшится до нормального уровня. В сетях X.25 такой протокол необходим, т. к. эти сети никогда не использовали разделяемые среды передачи данных, а работали по индивидуальным каналам связи в полнодуплексном режиме.

При разработке коммутаторов локальных сетей ситуация коренным образом отличалась от ситуации, при которой создавались коммутаторы территориальных сетей. Основной задачей было сохранение конечных узлов в неизменном виде, что исключало корректировку протоколов локальных сетей. А в этих протоколах процедур управления потоком не было – общая среда передачи данных в режиме разделения времени исключала возникновение ситуаций, когда сеть переполнялась бы необработанными кадрами. Сеть не накапливала данных в каких-либо промежуточных буферах при использовании только повторителей или концентраторов.

В марте 1997 года принят стандарт IEEE 802.3x на управление потоком в полнодуплексных версиях протокола Ethernet. Он определяет весьма простую процедуру управления потоком, подобную той, которая используется в протоколах LLC2 и LAP-B. Эта процедура подразумевает две команды – «Приостановить передачу» и «Возобновить передачу», которые направляются соседнему узлу. Отличие от протоколов типа LLC2 в том, что эти команды реализуются на уровне символов кодов физического уровня, таких как 4B/5B, а не на уровне команд, оформленных в специальные управляющие кадры. Сетевой адаптер или порт коммутатора, поддерживающий стандарт 802.3x и получивший команду «Приостановить передачу», должен прекратить передавать кадры впредь до получения команды «Возобновить передачу».

Проблема управления потоком кадров при полудуплексной работе

При работе порта в полудуплексном режиме коммутатор не может изменять протокол и пользоваться для управления потоком новыми командами, такими как «Приостановить передачу» и «Возобновить передачу». Зато у коммутатора появляется возможность воздействовать на конечный узел с помощью механизмов алгоритма доступа к среде, который конечный узел обязан отрабатывать. Эти приемы основаны на том, что конечные узлы строго соблюдают все параметры алгоритма доступа к среде, а порты коммутатора – нет. Обычно применяются два основных способа управления потоком кадров: обратное давление на конечный узел и агрессивный захват среды.

Метод обратного давления (backpressure) состоит в создании искусственных коллизий в сегменте, который чересчур интенсивно посылает кадры в коммутатор. Для этого коммутатор обычно использует jam-последовательность, отправляемую на выход порта, к которому подключен сегмент (или узел), чтобы приостановить его активность. Кроме того, метод обратного давления может применяться в тех случаях, когда процессор порта не рассчитан на поддержку максимально возможного для данного протокола трафика.

Второй метод «торможения» конечного узла в условиях перегрузки внутренних буферов коммутатора основан на так называемом агрессивном поведении порта коммутатора при захвате среды либо после окончания передачи очередного пакета, либо после коллизии. Эти два случая иллюстрируются рис. 2.75, *а* и *б*.

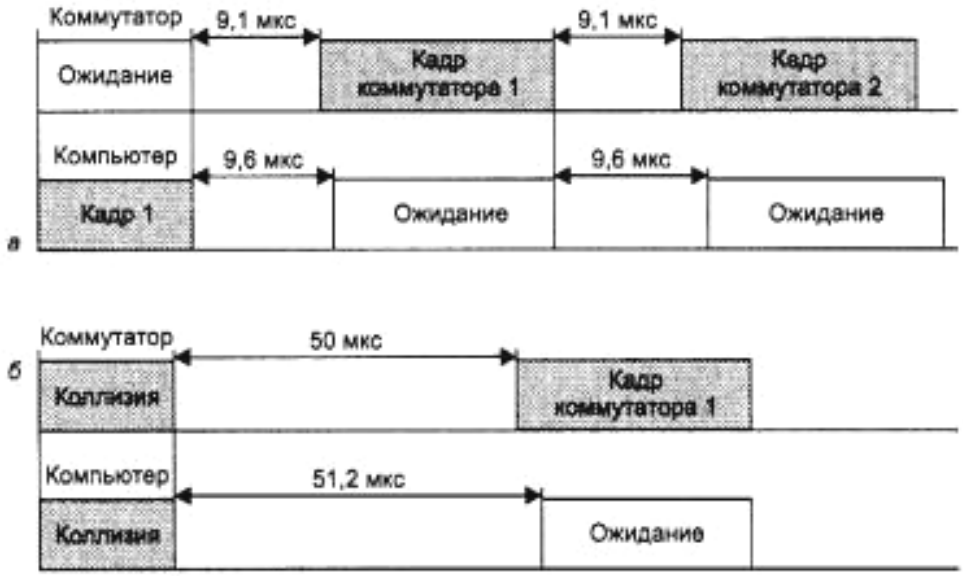


Рис. 2.75. Агрессивное поведение коммутатора при перегрузках буферов

В первом случае коммутатор окончил передачу очередного кадра и вместо технологической паузы в 9,6 мкс сделал паузу в 9,1 мкс и начал передачу нового кадра. Компьютер не смог захватить среду, т. к. он выдержал стандартную паузу в 9,6 мкс и обнаружил после этого, что среда уже занята.

Во втором случае кадры коммутатора и компьютера столкнулись и была зафиксирована коллизия. Так как компьютер сделал паузу после коллизии в 51,2 мкс, как это положено по стандарту (интервал отсрочки равен 512 битовых интервалов), а коммутатор – 50 мкс, то и в этом случае компьютеру не удалось передать свой кадр.

Коммутатор может пользоваться этим механизмом адаптивно, увеличивая степень своей агрессивности по мере необходимости.

Многие производители реализуют с помощью сочетания описанных двух методов достаточно тонкие механизмы управления потоком кадров при перегрузках. Эти методы используют алгоритмы чередования передаваемых и принимаемых кадров (*frame interleave*). Алгоритм чередования должен быть гибким и позволять компьютеру в критических ситуациях на каждый принимаемый кадр передавать несколько своих, разгружая внутренний буфер кадров, причем не обязательно снижая при этом интенсивность приема кадров до нуля, а просто уменьшая ее до необходимого уровня.

2.23. Техническая реализация коммутаторов

Несмотря на то, что в коммутаторах работают известные и хорошо отработанные алгоритмы прозрачных мостов и мостов с маршрутизацией от источника, существует большое разнообразие моделей коммутаторов.

В настоящее время коммутаторы используют в качестве базовой одну из трех схем, на которой строится такой узел обмена:

- коммутационная матрица;
- разделяемая многовходовая память;
- общая шина.

Часто эти три способа взаимодействия комбинируются в одном коммутаторе.

Коммутаторы на основе коммутационной матрицы

Коммутационная матрица обеспечивает основной и самый быстрый способ взаимодействия процессоров портов, именно он был реализован в первом промышленном коммутаторе локальных сетей. Однако реализация

матрицы возможна только для определенного числа портов, причем сложность схемы возрастает пропорционально квадрату количества портов коммутатора (рис. 2.76).

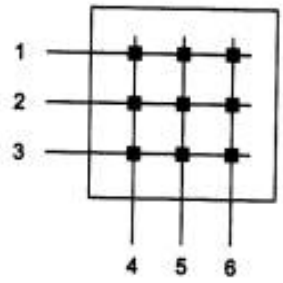


Рис. 2.76. Коммутационная матрица

Более детальное представление одного из возможных вариантов реализации коммутационной матрицы для 8-и портов дано на рис. 2.77. Входные блоки процессоров портов на основании просмотра адресной таблицы коммутатора определяют по адресу назначения номер выходного порта. Эту информацию они добавляют к байтам исходного кадра в виде специального ярлыка – тэга (*tag*). Для данного примера тэг представляет собой просто 3-хразрядное двоичное число, соответствующее номеру выходного порта.

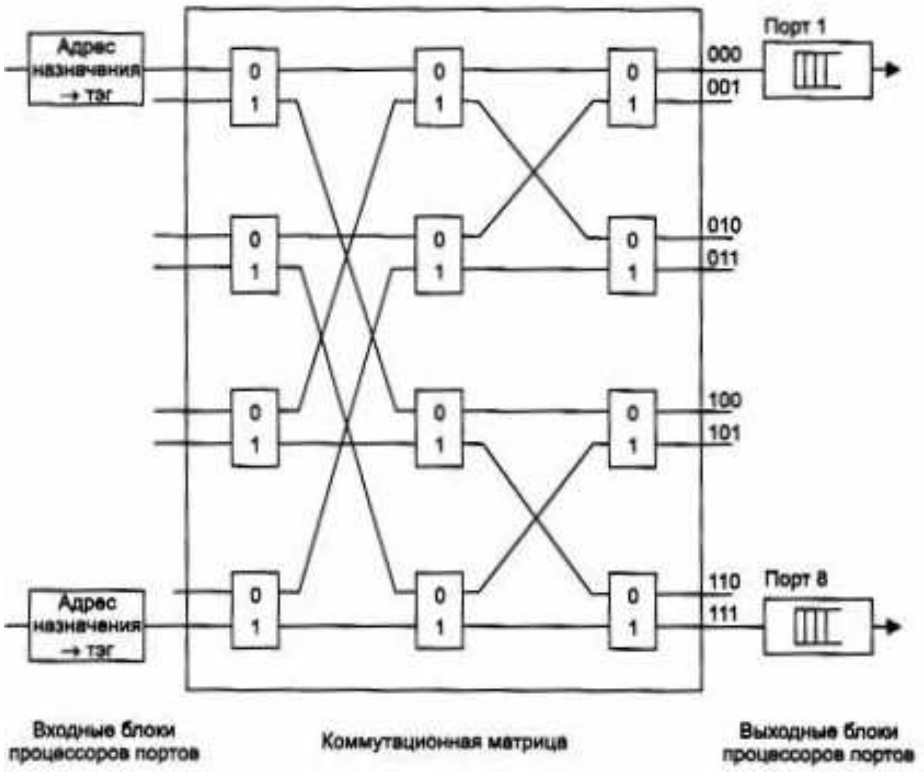


Рис. 2.77. Реализация коммутационной матрицы 8×8
с помощью двоичных переключателей

Матрица состоит из трех уровней двоичных переключателей, которые соединяют свой вход с одним из двух выходов в зависимости от значения бита тэга. Переключатели первого уровня управляются первым битом тэга, второго – вторым, а третьего – третьим.

Матрица может быть реализована и по-другому, на основании комбинационных схем другого типа, но ее особенностью все равно остается технология коммутации физических каналов. Известным недостатком этой технологии является отсутствие буферизации данных внутри коммутационной матрицы – если составной канал невозможно построить из-за занятости выходного порта или промежуточного коммутационного элемента, то данные должны накапливаться в их источнике, в данном случае – во входном блоке порта, принявшего кадр. Основные достоинства таких матриц – высокая скорость коммутации и регулярная структура, которую удобно реализовывать в интегральных микросхемах. Зато после реализации матрицы N×N в составе БИС проявляется еще один ее недостаток – сложность наращивания числа коммутируемых портов.

Коммутаторы с общей шиной

В коммутаторах с общей шиной процессоры портов связывают высокоскоростной шиной, используемой в режиме разделения времени.

Пример такой архитектуры приведен на рис. 2.78. Чтобы шина не блокировала работу коммутатора, ее производительность должна равняться, по крайней мере, сумме производительностей всех портов коммутатора. Для модульных коммутаторов некоторые сочетания модулей с низкоскоростными портами могут приводить к неблокирующей работе, а установка модулей с высокоскоростными портами может приводить к тому, что блокирующим элементом станет, например, общая шина.

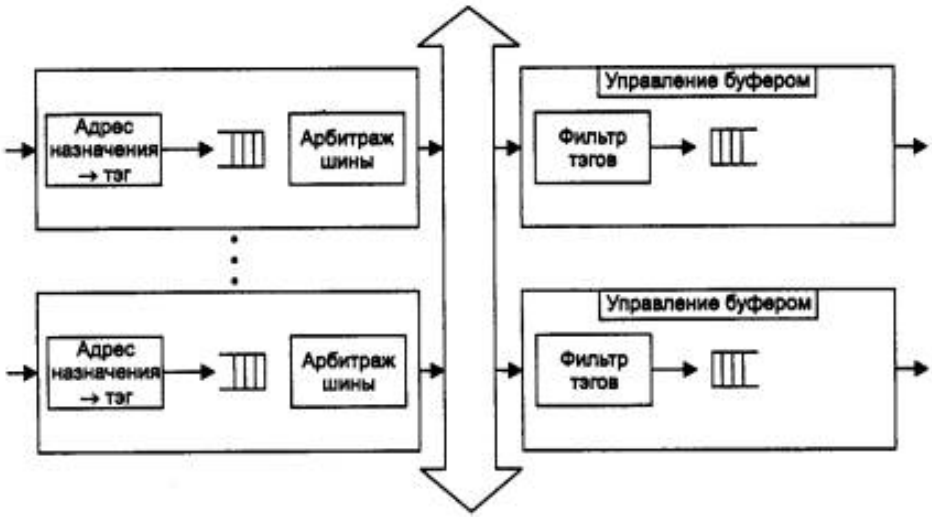


Рис. 2.78. Архитектура коммутатора с общей шиной

Кадр должен передаваться по шине небольшими частями, по нескольку байт, чтобы передача кадров между несколькими портами происходила в псевдопараллельном режиме, не внося задержек в передачу кадра в целом. Размер такой ячейки данных определяется производителем коммутатора. Некоторые производители, например LANNET или Centillion, выбрали в качестве порции данных, переносимых за одну операцию по шине, ячейку ATM с ее полем данных в 48 байт. Такой подход облегчает трансляцию протоколов локальных сетей в протокол ATM, если коммутатор поддерживает эти технологии.

Входной блок процессора помещает в ячейку, переносимую по шине, тэг, в котором указывает номер порта назначения. Каждый выходной блок процессора порта содержит фильтр тэгов, который выбирает тэги, предназначенные данному порту.

Шина, так же как и коммутационная матрица, не может осуществлять промежуточную буферизацию, но так как данные кадра разбиваются на небольшие ячейки, то задержек с начальным ожиданием доступности выходного порта в такой схеме нет – здесь работает принцип коммутации пакетов, а не каналов.

Коммутаторы с разделяемой памятью

Третья базовая архитектура взаимодействия портов – двухвходовая разделяемая память. Пример такой архитектуры приведен на рис. 2.79.

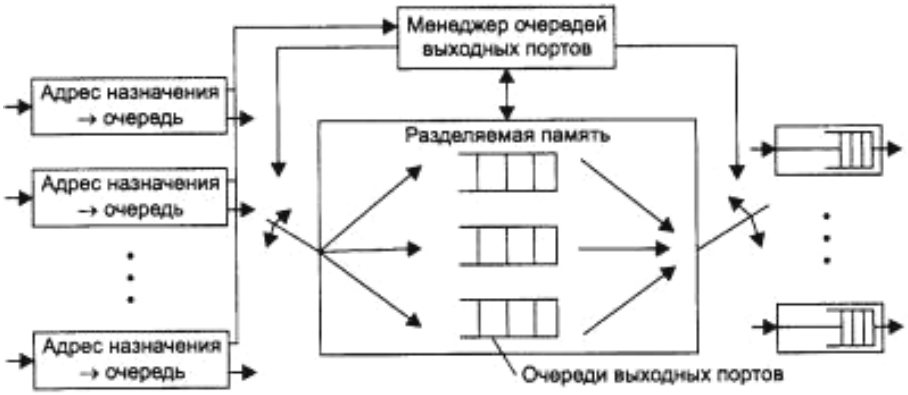


Рис. 2.79. Архитектура разделяемой памяти

Входные блоки процессоров портов соединяются с переключаемым входом разделяемой памяти, а выходные блоки этих же процессоров соединяются с переключаемым выходом этой памяти. Переключением входа и выхода разделяемой памяти управляет менеджер очередей выходных портов. В разделяемой памяти менеджер организует несколько очередей данных, по одной для каждого выходного порта. Входные блоки процессо-

ров передают менеджеру портов запросы на запись данных в очередь того порта, который соответствует адресу назначения пакета. Менеджер по очереди подключает вход памяти к одному из входных блоков процессоров, и тот переписывает часть данных кадра в очередь определенного выходного порта. По мере заполнения очередей менеджер производит также поочередное подключение выхода разделяемой памяти к выходным блокам процессоров портов, и данные из очереди переписываются в выходной буфер процессора.

Память должна быть достаточно быстродействующей для поддержания скорости переписи данных между N портами коммутатора. Применение общей буферной памяти, гибко распределяемой менеджером между отдельными портами, снижает требования к размеру буферной памяти процессора порта.

Комбинированные коммутаторы

У каждой из описанных архитектур есть свои преимущества и недостатки, поэтому часто в сложных коммутаторах эти архитектуры применяются в комбинации друг с другом. Пример такого комбинирования приведен на рис. 2.80.

Рис. 2.80. Комбинирование архитектур коммутационной матрицы и общей шины

Коммутатор состоит из модулей с фиксированным количеством портов (2 – 12), выполненных на основе специализированной БИС, реализующей архитектуру коммутационной матрицы. Если порты, между которыми нужно передать кадр данных, принадлежат одному модулю, то передача кадра осуществляется процессорами модуля на основе имеющейся в модуле коммутационной матрицы. Если же порты принадлежат разным модулям, то процессоры общаются по общей шине. При такой архитектуре передача кадров внутри модуля будет происходить быстрее, чем при межмодульной передаче, т. к. коммутационная матрица – наиболее быстрый, хотя и наименее масштабируемый способ взаимодействия портов. Скорость

внутренней шины коммутаторов может достигать нескольких Гбит/с, а у наиболее мощных моделей – до 20 – 30 Гбит/с.

Можно представить и другие способы комбинирования архитектур, например, использование разделяемой памяти для взаимодействия модулей.

Конструктивное исполнение коммутаторов

В конструктивном отношении коммутаторы делятся на следующие типы:

- автономные коммутаторы с фиксированным количеством портов;
- модульные коммутаторы на основе шасси;
- коммутаторы с фиксированным количеством портов, собираемые в стек.

Первый тип коммутаторов обычно предназначен для организации небольших рабочих групп.

Модульные коммутаторы на основе шасси чаще всего предназначены для применения на магистрали сети. Поэтому они выполняются на основе какой-либо комбинированной схемы, в которой взаимодействие модулей организуется по быстродействующей шине или же на основе быстрой разделяемой памяти большого объема. Модули такого коммутатора выполняются на основе технологии «hot swap», т. е. допускают замену на ходу, без выключения коммутатора, т. к. центральное коммуникационное устройство сети не должно иметь перерывов в работе. Шасси обычно снабжается резервированными источниками питания и резервированными вентиляторами в тех же целях.

С технической точки зрения определенный интерес представляют стековые коммутаторы. Эти устройства представляют собой коммутаторы, которые могут работать автономно, т. к. выполнены в отдельном корпусе, но имеют специальные интерфейсы, которые позволяют их объединять в общую систему, работающую как единый коммутатор. Говорят, что в этом случае отдельные коммутаторы образуют стек.

Обычно такой специальный интерфейс представляет собой высокоскоростную шину, которая позволяет объединить отдельные корпуса подобно модулям в коммутаторе на основе шасси. Так как расстояния между корпусами больше, чем между модулями на шасси, скорость обмена по шине обычно ниже, чем у модульных коммутаторов: 200 – 400 Мбит/с. Не очень высокие скорости обмена между коммутаторами стека обусловлены также тем, что стековые коммутаторы обычно занимают промежуточное

положение между коммутаторами с фиксированным количеством портов и коммутаторами на основе шасси. Стековые коммутаторы применяются для создания сетей рабочих групп и отделов, поэтому сверхвысокие скорости шин обмена им не очень нужны и не соответствуют их ценовому диапазону.

Структура стека коммутаторов, соединяемых по скоростным специальным портам, показана на рис. 2.81.

Существуют коммутаторы, которые позволяют объединить два коммутатора полнодуплексным каналом более чем по одной паре портов. Например, коммутаторы модели 28115 компании Nortel Networks имеют по два порта Fast Ethernet, с помощью которых можно соединять коммутаторы, образуя полнодуплексный канал с производительностью 400 Мбит/с (рис. 2.82).

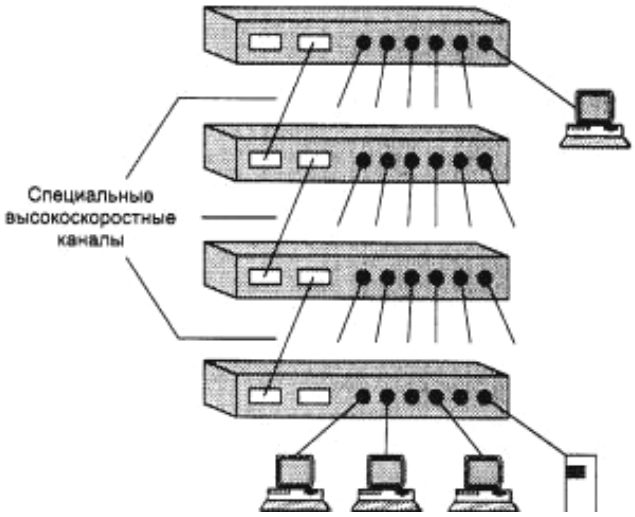


Рис. 2.81. Стек коммутаторов, объединяемых по высокоскоростным каналам

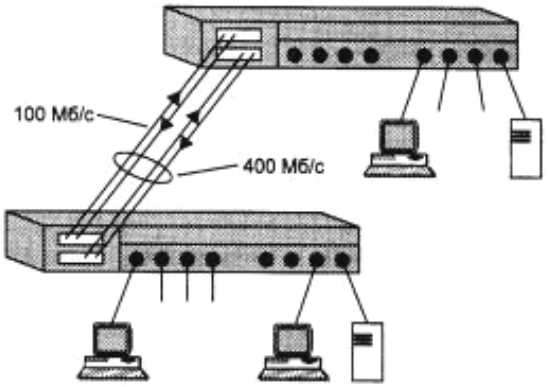


Рис. 2.82. Транковое полнодуплексное соединение коммутаторов 28115 компании Nortel Networks

Такие соединения называются *транковыми* и являются частной разработкой каждой компании, выпускающей коммуникационное оборудование, т. к. нарушают не только логику доступа к разделяемым средам, но и топологию соединения мостов, запрещающую петлевидные контуры (а такой контур всегда образуется при соединении коммутаторов более чем одной парой портов). При соединении коммутаторов разных производителей транк работать не будет, т. к. каждый производитель добавляет к логике изучения адресов сети коммутатором по транковой связи что-то свое, чтобы добиться от него правильной работы.

2.24. Характеристики, влияющие на производительность коммутаторов, дополнительные функции

Характеристики коммутаторов

Производительность коммутатора – то свойство, которое интересует сетевых интеграторов и администратов в этом устройстве в первую очередь.

Основными показателями коммутатора, характеризующими его производительность, являются:

- скорость фильтрации кадров;
- скорость продвижения кадров;
- пропускная способность;
- задержка передачи кадра.

Кроме того, существует несколько характеристик коммутатора, которые в наибольшей степени влияют на указанные характеристики производительности. К ним относятся:

- тип коммутации – «на лету» или с полной буферизацией;
- размер буфера (буферов) кадров;
- производительность внутренней шины;
- производительность процессора или процессоров;
- размер внутренней адресной таблицы.

Скорость фильтрации (filtering) 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, т. к. его порт назначения и порт источника принадлежат одному логическому сегменту.

Скорость фильтрации практически у всех коммутаторов является неблокирующей – коммутатор успевает отбрасывать кадры в темпе их поступления.

Скорость продвижения (forwarding) 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 для какого размера кадра приведены значения скоростей фильтрации и продвижения, то по умолчанию считается, что эти показатели даются для протокола Ethernet и кадров минимального размера, т. е. кадров длиной 64 байта (без преамбулы) с полем данных в 46 байт. Если скорости указаны для какого-либо определенного протокола, например Token Ring или FDDI, то они также даны для кадров минимальной длины этого протокола (например, кадров длины 29 байт для протокола FDDI). Применение 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 способность коммутатора работать при наихудшем сочетании параметров трафика. Кроме того, для пакетов минимальной длины скорость фильтрации и продвижения максимальна, что имеет немаловажное значение при рекламе коммутатора.

Пропускная способность коммутатора измеряется количеством пользовательских данных (в мегабитах в секунду), переданных в единицу времени через его порты. Так как коммутатор работает на канальном уровне, 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, т. к. при этом доля накладных расходов на служебную информацию кадра гораздо ниже, чем для кадров минимальной длины, а время выполнения коммута-

тором операций по обработке кадра, приходящееся на один байт пользовательской информации, существенно меньше. Поэтому коммутатор может быть блокирующим для кадров минимальной длины, но при этом иметь очень хорошие показатели пропускной способности.

Задержка передачи кадра измеряется как время, прошедшее с момента прихода первого байта кадра на входной порт коммутатора до момента появления этого байта на его выходном порту. Задержка складывается из времени, затрачиваемого на буферизацию байт кадра, а также времени, затрачиваемого на обработку кадра коммутатором, – просмотра адресной таблицы, принятия 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 5 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. Обычно производители коммутаторов указывают общую максимальную пропускную способность устройства.

Размер адресной таблицы. Максимальная емкость адресной таблицы определяет предельное количество MAC-адресов, 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. Экземпляры адресной таблицы разных процессорных модулей не обязательно содержат одну и ту же адресную информацию – скорее всего, повторяющихся адресов будет не так много, если только распределение трафика каждого порта между остальными портами не полностью равновероятно. Каждый порт хранит только те наборы адресов, с которыми он работал в последнее время.

Значение максимального числа MAC-адресов, которые может запомнить процессор порта, зависит от области применения коммутатора. Коммутаторы рабочих групп обычно поддерживают всего несколько адресов на порт, т. к. они предназначены для образования микросегментов. Комму-

таторы отделов должны поддерживать несколько сотен адресов, а коммутаторы магистралей сетей – до нескольких тысяч, обычно 4000 – 8000 адресов.

Недостаточная емкость адресной таблицы может служить причиной замедления работы коммутатора и засорения сети избыточным трафиком. Если адресная таблица процессора порта полностью заполнена, а он встречает новый адрес источника в поступившем пакете, 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 будет создавать лишнюю работу для многих процессоров портов, кроме того, копии этого кадра будут попадать и на 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 производится в расчете на то, что этот порт подключен к вышестоящему коммутатору при иерархическом соединении коммутаторов в крупной сети, который имеет достаточную емкость адресной таблицы и знает, куда нужно передать любой кадр.

Объем буфера кадров. Внутренняя буферная память коммутатора нужна для временного хранения кадров данных в тех 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, это не гарантирует, что их производительности хватит при пиковых значениях нагрузок. 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 50 – 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 переполнится.

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 можно перераспределять между несколькими портами, т. к. 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.

Дополнительные функции коммутаторов

Так как коммутатор представляет собой сложное вычислительное устройство, имеющее несколько процессорных модулей, то целесообразно нагрузить его помимо выполнения основной функции передачи кадров с порта на порт некоторыми дополнительными функциями, полезными при построении надежных и гибких сетей. Ниже описываются наиболее распространенные дополнительные функции коммутаторов, которые поддерживаются большинством производителей коммуникационного оборудования.

Поддержка алгоритма Spanning Tree. Алгоритм покрывающего дерева – Spanning Tree Algorithm (STA) – позволяет коммутаторам автоматически определять древовидную конфигурацию связей в сети при произвольном соединении портов между собой. Как уже отмечалось ранее, для нормальной работы коммутатора требуется отсутствие замкнутых маршрутов в сети. Эти маршруты могут создаваться администратором специально для образования резервных связей или же возникать случайным образом, что вполне возможно, если сеть имеет многочисленные связи, а кабельная система плохо структурирована или документирована.

Поддерживающие алгоритм STA коммутаторы автоматически создают активную древовидную конфигурацию связей (т. е. связную конфигурацию без петель) на множестве всех связей сети. Такая конфигурация

называется покрывающим деревом – Spanning Tree (иногда ее называют основным деревом), и ее название дало имя всему алгоритму. Алгоритм Spanning Tree описан в стандарте IEEE 802.1D, том же стандарте, который определяет принципы работы прозрачных мостов.

Коммутаторы находят покрывающее дерево адаптивно, с помощью обмена служебными пакетами. Реализация в коммутаторе алгоритма STA очень важна для работы в больших сетях – если коммутатор не поддерживает этот алгоритм, то администратор должен самостоятельно определить, какие порты нужно перевести в заблокированное состояние, чтобы исключить петли. К тому же при отказе какого-либо кабеля, порта или коммутатора администратор должен, во-первых, обнаружить факт отказа, а, во-вторых, – ликвидировать последствия отказа, переведя резервную связь в рабочий режим путем активизации некоторых портов. При поддержке коммутаторами сети протокола Spanning Tree отказы обнаруживаются автоматически, за счет постоянного тестирования связности сети служебными пакетами. После обнаружения потери связности протокол строит новое покрывающее дерево, если это возможно, и сеть автоматически восстанавливает работоспособность.

Алгоритм Spanning Tree определяет активную конфигурацию сети за три этапа.

На первом этапе в сети определяется корневой коммутатор (*root switch*), от которого строится дерево. Корневой коммутатор может быть выбран автоматически или назначен администратором. При автоматическом выборе корневым становится коммутатор с меньшим значением MAC-адреса его блока управления.

На втором этапе для каждого коммутатора определяется корневой порт (*root port*) – это порт, который имеет по сети кратчайшее расстояние до корневого коммутатора (точнее, до любого из портов корневого коммутатора).

На третьем этапе для каждого сегмента сети выбирается так называемый назначенный порт (*designated port*) – это порт, который имеет кратчайшее расстояние от данного сегмента до корневого коммутатора. После определения корневых и назначенных портов каждый коммутатор блокирует остальные порты, которые не попали в эти два класса портов. При таком выборе активных портов в сети исключаются петли и оставшиеся связи образуют покрывающее дерево (если оно может быть построено при существующих связях в сети).

Понятие расстояния играет важную роль в построении покрывающего дерева. Именно по этому критерию выбирается единственный порт, со-

единяющий каждый коммутатор с корневым коммутатором, и единственный порт, соединяющий каждый сегмент сети с корневым коммутатором.

На рис. 2.83 показан пример построения конфигурации покрывающего дерева для сети, состоящей из 5-и сегментов и 5-и коммутаторов. Корневые порты закрашены темным цветом, назначенные порты не закрашены, а заблокированные порты перечеркнуты. В активной конфигурации коммутаторы 2 и 4 не имеют портов, передающих кадры данных, поэтому они закрашены как резервные.

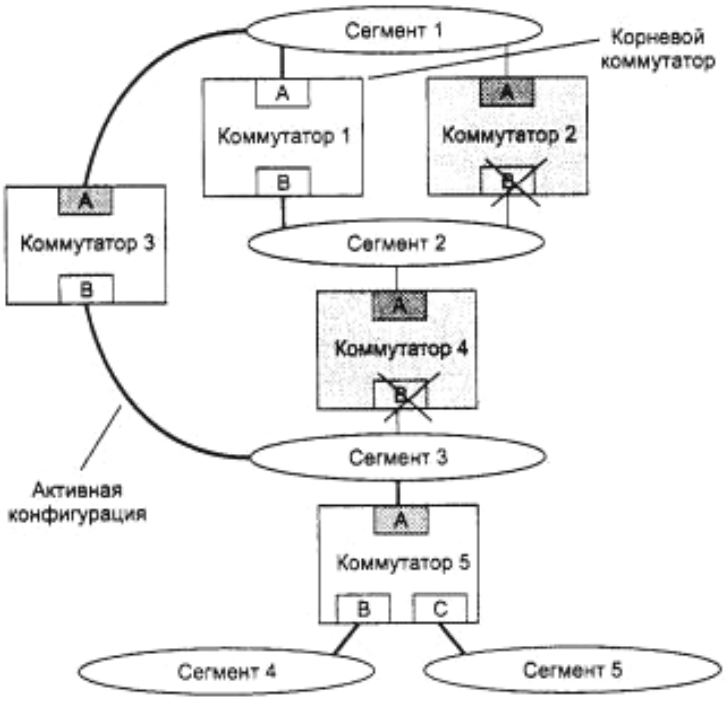


Рис. 2.83. Построение покрывающего дерева сети по алгоритму STA

Расстояние до корня определяется как суммарное условное время на передачу одного бита данных от порта данного коммутатора до порта корневого коммутатора. При этом считается, что время внутренних передач данных (с порта на порт) коммутатором пренебрежимо мало, а учитывается только время на передачу данных по сегментам сети, соединяющим коммутаторы. Условное время сегмента рассчитывается как время, затрачиваемое на передачу одного бита информации в 10-и наносекундных единицах между непосредственно связанными по сегменту сети портами.

В приведенном примере предполагается, что все сегменты работают на одной скорости, поэтому они имеют одинаковые условные расстояния, которые поэтому не показаны на рисунке.

Для автоматического определения начальной активной конфигурации дерева все коммутаторы сети после их инициализации начинают пе-

риодически обмениваться специальными пакетами, называемыми *протокольными блоками данных моста – BPDU (Bridge Protocol Data Unit)*, что отражает факт первоначальной разработки алгоритма STA для мостов.

Пакеты BPDU помещаются в поле данных кадров канального уровня, например кадров Ethernet или FDDI. Желательно, чтобы все коммутаторы поддерживали общий групповой адрес, с помощью которого кадры, содержащие пакеты BPDU, могли бы одновременно передаваться всем коммутаторам сети. Иначе пакеты BPDU рассылаются широковещательно.

Идентификаторы коммутаторов состоят из 8 байт, причем младшие 6 являются MAC-адресом блока управления коммутатора. Старшие 2 байта в исходном состоянии заполнены нулями, но администратор может изменить значение этих байтов, тем самым назначив определенный коммутатор корневым.

После инициализации каждый коммутатор сначала считает себя корневым. Поэтому он начинает через определенный интервал генерировать через все свои порты сообщения BPDU конфигурационного типа. В них он указывает свой идентификатор в качестве идентификатора корневого коммутатора (и в качестве идентификатора данного коммутатора также), расстояние до корня устанавливается в 0, а в качестве идентификатора порта указывается идентификатор того порта, через который передается BPDU. Как только коммутатор получает BPDU, в котором имеется идентификатор корневого коммутатора, со значением, меньшим его собственного, он перестает генерировать свои собственные кадры BPDU, а начинает ретранслировать только кадры нового претендента на звание корневого коммутатора. На рис. 2.83 у коммутатора 1 идентификатор имеет наименьшее значение, т. к. он стал в результате обмена кадрами корневым.

При ретрансляции кадров каждый коммутатор наращивает расстояние до корня, указанное в пришедшем BPDU, на условное время сегмента, по которому принят данный кадр. Тем самым в кадре BPDU, по мере прохождения через коммутаторы, накапливается расстояние до корневого коммутатора. Если считать, что все сегменты рассматриваемого примера являются сегментами Ethernet, то коммутатор 2, приняв от коммутатора BPDU по сегменту 1 с расстоянием, равным 0, наращивает его на 10 единиц.

Ретранслируя кадры, каждый коммутатор для каждого своего порта запоминает минимальное расстояние до корня, встретившееся во всех принятых этим портом кадрах BPDU. При завершении процедуры установления конфигурации покрывающего дерева (по времени) каждый коммутатор находит свой корневой порт – это порт, для которого минимальное

расстояние до корня оказалось меньше, чем у других портов. Так, коммутатор 3 выбирает порт А в качестве корневого, поскольку по порту А минимальное расстояние до корня равно 10 (BPDU с таким расстоянием принят от корневого коммутатора через сегмент 1). Порт В коммутатора 3 обнаружил в принимаемых кадрах минимальное расстояние в 20 единиц – это соответствовало случаю прохождения кадра от порта В корневого моста через сегмент 2, затем через мост 4 и сегмент 3.

Кроме корневого порта коммутаторы распределенным образом выбирают для каждого сегмента сети назначенный порт. Для этого они исключают из рассмотрения свой корневой порт (для сегмента, к которому он подключен, всегда существует другой коммутатор, который ближе расположен к корню), а для всех своих оставшихся портов сравнивают принятые по ним минимальные расстояния до корня с расстоянием до корня своего корневого порта. Если у какого-либо своего порта принятые им расстояния до корня больше, чем расстояние маршрута, пролегающего через свой корневой порт, то это значит, что для сегмента, к которому подключен данный порт, кратчайшее расстояние к корневому коммутатору ведет именно через данный порт. Коммутатор делает все свои порты, у которых такое условие выполняется, назначенными.

Если в процессе выбора корневого порта или назначенного порта несколько портов оказываются равными по критерию кратчайшего расстояния до корневого коммутатора, то выбирается порт с наименьшим идентификатором.

В качестве примера рассмотрим выбор корневого порта для коммутатора 2 и назначенного порта для сегмента 2. Мост 2 при выборе корневого порта столкнулся с ситуацией, когда порт А и порт В имеют равное расстояние до корня – по 10 единиц (порт А принимает кадры от порта В корневого коммутатора через один промежуточный сегмент – сегмент 1, а порт В принимает кадры от порта А корневого коммутатора также через один промежуточный сегмент – через сегмент 2). Идентификатор А имеет меньшее числовое значение, чем В (в силу упорядоченности кодов символов), поэтому порт А стал корневым портом коммутатора 2.

При проверке порта В на случай, не является ли он назначенным для сегмента 2, коммутатор 2 обнаружил, что через этот порт он принимал кадры с указанным в них минимальным расстоянием 0 (это были кадры от порта В корневого коммутатора 1). Так как собственный корневой порт у коммутатора 2 имеет расстояние до корня 10, то порт В не является назначенным для сегмента 2.

Затем все порты, кроме корневого и назначенных, переводятся каждым коммутатором в заблокированное состояние. На этом построение покрывающего дерева заканчивается.

В процессе нормальной работы корневой коммутатор продолжает генерировать служебные кадры BPDU, а остальные коммутаторы продолжают их принимать своими корневыми портами и ретранслировать назначенными. Если у коммутатора нет назначенных портов, как у коммутаторов 2 и 4, то они все равно продолжают принимать участие в работе протокола Spanning Tree, принимая служебные кадры корневым портом. Если по истечении тайм-аута корневой порт любого коммутатора сети не получает служебный кадр BPDU, то он инициализирует новую процедуру построения покрывающего дерева, оповещая об этом другие коммутаторы BPDU уведомлением о реконфигурации. Получив такой кадр, все коммутаторы начинают снова генерировать BDPU конфигурационного типа, в результате чего устанавливается новая активная конфигурация.

Трансляция протоколов канального уровня. Коммутаторы могут выполнять трансляцию одного протокола канального уровня в другой, например, Ethernet в FDDI, Fast Ethernet в Token Ring и т. п. При этом они работают по тем же алгоритмам, что и транслирующие мосты.

Трансляцию протоколов локальных сетей облегчает тот факт, что наиболее сложную работу, которую при объединении гетерогенных сетей часто выполняют маршрутизаторы и шлюзы, а именно работу по трансляции адресной информации, в данном случае выполнять не нужно. Все конечные узлы локальных сетей имеют уникальные адреса одного и того же формата независимо от поддерживаемого протокола. Поэтому адрес сетевого адаптера Ethernet понятен сетевому адаптеру FDDI, и они могут использовать эти адреса в полях своих кадров, не задумываясь о том, что узел, с которым они взаимодействуют, принадлежит сети, работающей по другой технологии. Поэтому при согласовании протоколов локальных сетей коммутаторы не строят таблиц соответствия адресов узлов, а переносят адреса назначения и источника из кадра одного протокола в кадр другого.

Возможности коммутаторов по фильтрации трафика. Многие коммутаторы позволяют администраторам задавать дополнительные условия фильтрации кадров наряду со стандартными условиями их фильтрации в соответствии с информацией адресной таблицы. Пользовательские фильтры предназначены для создания дополнительных барьеров на пути кадров, которые ограничивают доступ определенных групп пользователей к определенным службам сети.

Наиболее простыми являются пользовательские фильтры на основе MAC-адресов станций. Так как MAC-адреса – это та информация, с которой работает коммутатор, то он позволяет задавать такие фильтры в удобной для администратора форме, например, отбрасывать кадры с определенным адресом. При этом пользователю, работающему на компьютере с данным MAC-адресом, полностью запрещается доступ к ресурсам другого сегмента сети.

Часто администратору требуется задать более тонкие условия фильтрации, например, запретить некоторому пользователю печатать свои документы на определенном сервере печати NetWare чужого сегмента, а остальные ресурсы этого сегмента сделать доступными. Для реализации такого фильтра нужно запретить передачу кадров с определенным MAC-адресом, в которых вложены пакеты IPX, в поле «номер сокета» которых будет указано значение, соответствующее службе печати NetWare. Коммутаторы не анализируют протоколы верхних уровней, такие как IPX, поэтому администратору приходится для задания условий такой фильтрации вручную определять поле, по значению которого нужно осуществлять фильтрацию, в виде пары «смещение – размер» относительно начала поля данных кадра канального уровня, а затем еще указать в шестнадцатеричном формате значение этого поля для службы печати.

Обычно условия фильтрации записываются в виде булевых выражений, формируемых с помощью логических операторов AND и OR.

Наложение дополнительных условий фильтрации может снизить производительность коммутатора, т. к. вычисление булевых выражений требует проведения дополнительных вычислений процессорами портов.

Приоритетная обработка кадров. Построение сетей на основе коммутаторов позволяет использовать приоритезацию трафика, причем делать это независимо от технологии сети. Эта новая возможность (по сравнению с сетями, построенными целиком на концентраторах) является следствием того, что коммутаторы буферизуют кадры перед их отправкой на другой порт. Коммутатор обычно ведет для каждого входного и выходного порта не одну, а несколько очередей, причем каждая очередь имеет свой приоритет обработки. При этом коммутатор может быть сконфигурирован, например, так, чтобы передавать один низкоприоритетный пакет на каждые 10 высокоприоритетных пакетов.

Поддержка приоритетной обработки может особенно пригодиться для приложений, предъявляющих различные требования к допустимым задержкам кадров и к пропускной способности сети для потока кадров.

Основным вопросом при приоритетной обработке кадров коммутаторами является вопрос назначения кадру приоритета. Так как не все протоколы канального уровня поддерживают поле приоритета кадра, например, у кадров Ethernet оно отсутствует, то коммутатор должен использовать какой-либо дополнительный механизм для связывания кадра с его приоритетом. Наиболее распространенный способ – приписывание приоритета портам коммутатора. При этом способе коммутатор помещает кадр в очередь кадров соответствующего приоритета в зависимости от того, через какой порт поступил кадр в коммутатор.

2.25. Виртуальные локальные сети. Типовые схемы применения коммутаторов в локальных сетях

Виртуальные локальные сети

Кроме своего основного назначения – повышения пропускной способности связей в сети – коммутатор позволяет локализовывать потоки информации в сети, а также контролировать эти потоки и управлять ими, опираясь на механизм пользовательских фильтров. Однако пользовательский фильтр может запретить передачи кадров только по конкретным адресам, а широковещательный трафик он передает всем сегментам сети. Так требует алгоритм работы моста, который реализован в коммутаторе, поэтому сети, созданные на основе мостов и коммутаторов, иногда называют плоскими – из-за отсутствия барьеров на пути широковещательного трафика.

Технология *виртуальных локальных сетей (Virtual LAN, VLAN)* позволяет преодолеть указанное ограничение. Виртуальной сетью называется группа узлов сети, трафик которой, в том числе и широковещательный, на канальном уровне полностью изолирован от других узлов сети (рис. 2.84). Это означает, что передача кадров между разными виртуальными сетями на основании адреса канального уровня невозможна, независимо от типа адреса – уникального, группового или широковещательного. В то же время внутри виртуальной сети кадры передаются по технологии коммутации, т. е. только на тот порт, который связан с адресом назначения кадра. Виртуальные сети могут пересекаться, если один или несколько компьютеров входят в состав более чем одной виртуальной сети. На рис. 2.84 сервер электронной почты входит в состав 3-й и 4-й виртуальных сетей. Это значит, что его кадры передаются коммутаторами всем компьютерам, входящим в эти сети. Если же какой-то компьютер входит в состав только виртуальной сети 3, то его кадры до сети 4 доходить не будут, но он может

взаимодействовать с компьютерами сети 4 через общий почтовый сервер. Такая схема не полностью защищает виртуальные сети друг от друга – так, широковещательный шторм, возникший на сервере электронной почты, захлестнет сеть 3 и сеть 4.

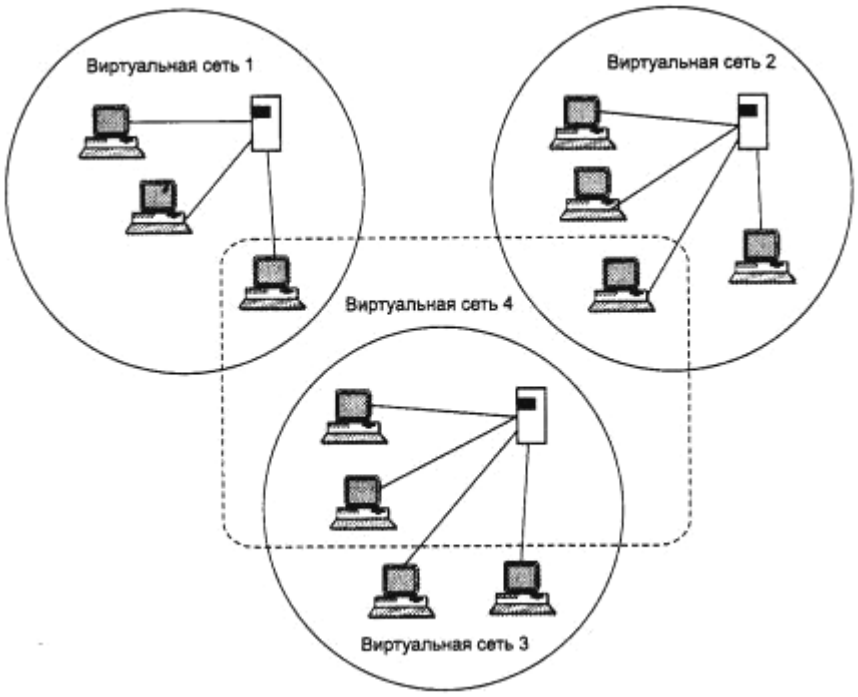


Рис. 2.84. Виртуальные сети

Говорят, что виртуальная сеть образует *домен широковещательного трафика (broadcast domain)*, по аналогии с доменом коллизий, который образуется повторителями сетей Ethernet.

Назначение технологии виртуальных сетей состоит в облегчении процесса создания изолированных сетей, которые затем должны связываться с помощью маршрутизаторов, реализующих какой-либо протокол сетевого уровня, например IP. Такое построение сети создает гораздо более мощные барьеры на пути ошибочного трафика из одной сети в другую. Сегодня считается, что любая крупная сеть должна включать маршрутизаторы, иначе потоки ошибочных кадров, например широковещательных, будут периодически затапливать всю сеть через прозрачные для них коммутаторы, приводя ее в неработоспособное состояние.

Технология виртуальных сетей создает гибкую основу для построения крупной сети, соединенной маршрутизаторами, т. к. коммутаторы позволяют создавать полностью изолированные сегменты программным путем, не прибегая к физической коммутации.

До появления технологии VLAN для создания отдельной сети использовались либо физически изолированные сегменты коаксиального кабеля, либо не связанные между собой сегменты, построенные на повторителях и мостах. Затем эти сети связывались маршрутизаторами в единую составную сеть (рис. 2.85).

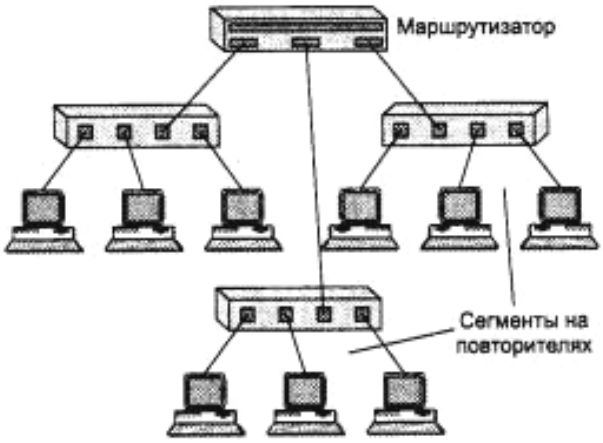


Рис. 2.85. Интерсеть, состоящая из сетей, построенных на основе повторителей

Изменение состава сегментов (переход пользователя в другую сеть, дробление крупных сегментов) при таком подходе подразумевает физическую перекоммутацию разъемов на передних панелях повторителей или в кроссовых панелях, что не очень удобно в больших сетях – много физической работы, к тому же высока вероятность ошибки.

Поэтому для устранения необходимости физической перекоммутации узлов стали применять многосегментные концентраторы, рассмотренные выше. Возникла возможность программировать состав разделяемого сегмента без физической перекоммутации.

Однако решение задачи изменения состава сегментов с помощью концентраторов накладывает большие ограничения на структуру сети – количество сегментов такого повторителя обычно невелико, поэтому выделить каждому узлу свой сегмент, как это можно сделать с помощью коммутатора, нереально. Кроме того, при таком подходе вся работа по передаче данных между сегментами ложится на маршрутизаторы. Поэтому сети, построенные на основе повторителей с конфигурационной коммутацией, по-прежнему основаны на разделении среды передачи данных между большим количеством узлов, и, следовательно, обладают гораздо меньшей производительностью по сравнению с сетями, построенными на основе коммутаторов.

При использовании технологии виртуальных сетей в коммутаторах одновременно решаются две задачи:

– повышение производительности в каждой из виртуальных сетей, т. к. коммутатор передает кадры в такой сети только узлу назначения;

– изоляция сетей друг от друга для управления правами доступа пользователей и создания защитных барьеров на пути широковещательных штормов.

Для связи виртуальных сетей в общую сеть требуется привлечение сетевого уровня. Он может быть реализован в отдельном маршрутизаторе, а может работать и в составе программного обеспечения коммутатора, который тогда становится комбинированным устройством – так называемым коммутатором 3-го уровня.

Технология образования и работы виртуальных сетей с помощью коммутаторов долгое время не стандартизировалась, хотя и была реализована в очень широком спектре моделей коммутаторов разных производителей. Такое положение изменилось после принятия в 1998 году стандарта IEEE 802.1Q, который определяет базовые правила построения виртуальных локальных сетей, не зависящие от протокола канального уровня, который поддерживает коммутатор.

При создании виртуальных сетей на основе одного коммутатора обычно используется механизм группирования в сети портов коммутатора (рис. 2.86).

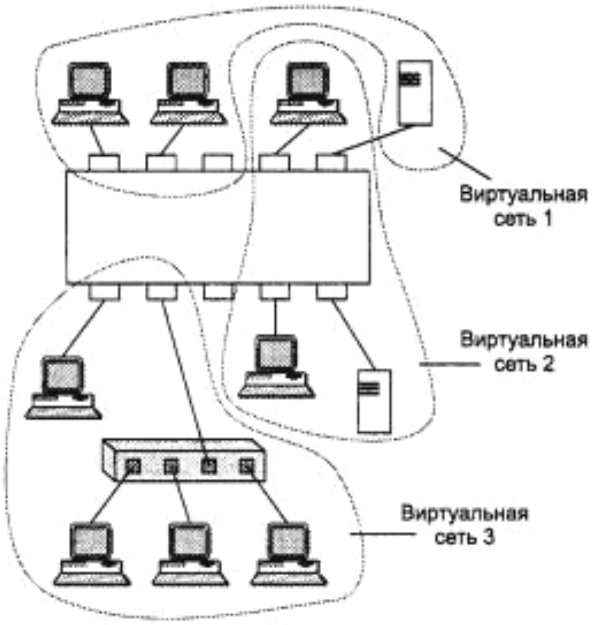


Рис. 2.86. Виртуальные сети, построенные на одном коммутаторе

При этом каждый порт приписывается той или иной виртуальной сети. Кадр, пришедший от порта, принадлежащего, например, виртуальной сети 1, никогда не будет передан порту, который не принадлежит этой виртуальной сети. Порт можно приписать нескольким виртуальным сетям, хотя на практике так делают редко – пропадает эффект полной изоляции сетей.

Группировка портов для одного коммутатора – наиболее логичный способ образования VLAN, т. к. виртуальных сетей, построенных на основе одного коммутатора, не может быть больше, чем портов. Если к одному порту подключен сегмент, построенный на основе повторителя, то узлы такого сегмента не имеет смысла включать в разные виртуальные сети – все равно трафик этих узлов будет общим.

Создание виртуальных сетей на основе группирования портов не требует от администратора большого объема ручной работы – достаточно каждый порт приписать к одной из нескольких заранее поименованных виртуальных сетей.

Второй способ образования виртуальных сетей основан на группировании MAC-адресов. Каждый MAC-адрес, который изучен коммутатором, приписывается той или иной виртуальной сети. При существовании в сети множества узлов этот способ требует выполнения большого количества ручных операций от администратора. Однако он оказывается более гибким при построении виртуальных сетей на основе нескольких коммутаторов, чем способ группирования портов.

Рисунок 2.87 иллюстрирует проблему, возникающую при создании виртуальных сетей на основе нескольких коммутаторов, поддерживающих технику группирования портов. Если узлы какой-либо виртуальной сети подключены к разным коммутаторам, то для соединения коммутаторов каждой такой сети должна быть выделена своя пара портов. В противном случае, если коммутаторы будут связаны только одной парой портов, информация о принадлежности кадра той или иной виртуальной сети при передаче из коммутатора в коммутатор будет утеряна. Таким образом, коммутаторы с группировкой портов требуют для своего соединения столько портов, сколько виртуальных сетей они поддерживают. Порты и кабели используются при таком способе очень расточительно. Кроме того, при соединении виртуальных сетей через маршрутизатор для каждой виртуальной сети выделяется в этом случае отдельный кабель и отдельный порт маршрутизатора, что также приводит к большим накладным расходам.

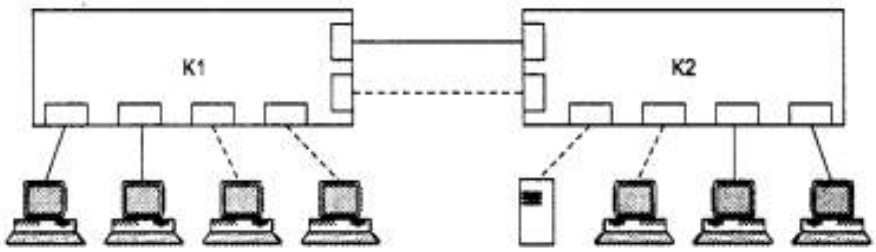


Рис. 2.87. Построение виртуальных сетей на нескольких коммутаторах с группировкой портов

Группирование MAC-адресов в виртуальную сеть на каждом коммутаторе избавляет от необходимости их связи несколькими портами, т. к. в этом случае MAC-адрес является меткой виртуальной сети. Однако этот способ требует выполнения большого количества ручных операций по маркировке MAC-адресов на каждом коммутаторе сети.

Описанные два подхода основаны только на добавлении дополнительной информации к адресным таблицам моста, и в них отсутствует возможность встраивания информации о принадлежности кадра к виртуальной сети в передаваемый кадр. Остальные подходы используют имеющиеся или дополнительные поля кадра для сохранения информации и принадлежности кадра при его перемещениях между коммутаторами сети. При этом нет необходимости запоминать в каждом коммутаторе принадлежность всех MAC-адресов интерсети виртуальным сетям.

Дополнительное поле с пометкой о номере виртуальной сети используется только тогда, когда кадр передается от коммутатора к коммутатору, а при передаче кадра конечному узлу оно удаляется. При этом модифицируется протокол взаимодействия «коммутатор – коммутатор», а программное и аппаратное обеспечение конечных узлов остается неизменным.

Для хранения номера виртуальной сети в стандарте IEEE 802.1Q предусмотрен тот же дополнительный заголовок, что и в стандарте 802.1p. Помимо 3-х бит для хранения приоритета кадра, описанных стандартом 802.1p, в этом заголовке 12 бит используются для хранения номера VLAN, к которой принадлежит кадр. Эта дополнительная информация позволяет коммутаторам разных производителей создавать до 4096 общих виртуальных сетей. Чтобы кадр Ethernet не увеличивался в объеме, при добавлении заголовка 802.1p/Q поле данных уменьшается на 2 байта.

Существуют два способа построения виртуальных сетей, которые используют уже имеющиеся поля для маркировки принадлежности кадра виртуальной сети, однако эти поля принадлежат не кадрам канальных протоколов, а пакетам сетевого уровня или ячейкам технологии ATM.

В первом случае виртуальные сети образуются на основе сетевых адресов, например адресов IP, т. е. той же информации, которая используется при построении интерсетей традиционным способом. Этот эффективный способ работает тогда, когда коммутаторы поддерживают не только протоколы канального уровня, но и протоколы сетевого уровня, т. е. являются комбинированными коммутаторами-маршрутизаторами, что бывает далеко не всегда.

Во втором случае виртуальные сети организуются с помощью виртуальных путей в АТМ-сетях.

Типовые схемы применения коммутаторов в локальных сетях

Сочетание коммутаторов и концентраторов. При построении небольших сетей, составляющих нижний уровень иерархии корпоративной сети, вопрос о применении того или иного коммуникационного устройства сводится к вопросу о выборе между концентратором или коммутатором.

При ответе на этот вопрос нужно принимать во внимание несколько факторов. Безусловно, немаловажное значение имеет стоимость в пересчете за порт, которую нужно заплатить при выборе устройства. Из технических соображений в первую очередь нужно принять во внимание существующее распределение трафика между узлами сети. Кроме того, нужно учитывать перспективы развития сети: будут ли в скором времени применяться мультимедийные приложения, будет ли модернизироваться компьютерная база. Если да, то нужно уже сегодня обеспечить резервы по пропускной способности применяемого коммуникационного оборудования. Использование технологии Intranet также ведет к увеличению объемов трафика, циркулирующего в сети, и это также необходимо учитывать при выборе устройства.

При выборе типа устройства – концентратор или коммутатор – нужно еще определить и тип протокола, который будут поддерживать его порты (или протоколов, если идет речь о коммутаторе, т. к. каждый порт может поддерживать отдельный протокол).

Сегодня выбор делается между протоколами трех скоростей – 10, 100 и 1000 Мбит/с. Поэтому, сравнивая применимость концентратора или коммутатора, необходимо рассмотреть варианты концентратора с портами на 10,100 и 1000 Мбит/с, а также несколько вариантов коммутаторов с различными комбинациями скоростей на портах.

Рассмотрим для примера вопрос о применимости коммутатора в сети с одним сервером и несколькими рабочими станциями, взаимодействующими только с сервером (рис. 2.88). Такая конфигурация сети часто встре-

чается в сетях масштаба рабочей группы, особенно в сетях NetWare, где стандартные клиентские оболочки не могут взаимодействовать друг с другом.

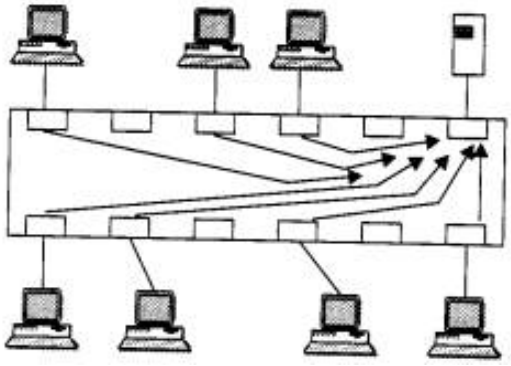


Рис. 2.88. Сеть с выделенным сервером

Если коммутатор имеет все порты с одинаковой пропускной способностью, например 10 Мбит/с, то пропускная способность порта в 10 Мбит/с будет распределяться между всеми компьютерами сети. Возможности коммутатора по повышению общей пропускной способности сети оказываются для такой конфигурации невостребованными. Несмотря на микросегментацию сети, ее пропускная способность ограничивается пропускной способностью протокола одного порта, как и в случае применения концентратора с портами 10 Мбит/с. Небольшой выигрыш при использовании коммутатора будет достигаться лишь за счет уменьшения количества коллизий – вместо коллизий кадры будут просто попадать в очередь к передатчику порта коммутатора, к которому подключен сервер.

Чтобы коммутатор работал в сетях с выделенным сервером более эффективно, производители коммутаторов выпускают модели с одним высокоскоростным портом на 100 Мбит/с для подключения сервера и несколькими низкоскоростными портами на 10 Мбит/с для подключения рабочих станций. В этом случае между рабочими станциями распределяется уже 100 Мбит/с, что позволяет обслуживать в неблокирующем режиме 10 – 30 станций в зависимости от интенсивности создаваемого ими трафика.

Однако с таким коммутатором может конкурировать концентратор, поддерживающий протокол с пропускной способностью 100 Мбит/с, например Fast Ethernet. Его стоимость в пересчете за порт будет несколько ниже стоимости за порт коммутатора с одним высокоскоростным портом, а производительность сети примерно та же.

Очевидно, что выбор коммуникационного устройства для сети с выделенным сервером достаточно сложен. Для принятия окончательного решения нужно принимать во внимание перспективы развития сети в отношении движения к сбалансированному трафику. Если в сети вскоре может появиться взаимодействие между рабочими станциями или же второй сервер, то выбор необходимо делать в пользу коммутатора, который сможет поддержать дополнительный трафик без ущерба по отношению к основному.

В пользу коммутатора может сыграть и фактор расстояний – применение коммутаторов не ограничивает максимальный диаметр сети величинами в 2500 м или 210 м, которые определяют размеры домена коллизий при использовании концентраторов Ethernet и Fast Ethernet.

В целом существует тенденция постепенного вытеснения концентраторов коммутаторами, которая наблюдается примерно с 1996 года.

Стянутая в точку магистраль на коммутаторе. При всем разнообразии структурных схем сетей, построенных на коммутаторах, все они используют две базовые структуры: стянутую в точку магистраль и распределенную магистраль. На основе этих базовых структур затем строятся разнообразные структуры конкретных сетей.

Стянутая в точку магистраль (collapsed backbone) – это структура, при которой объединение узлов, сегментов или сетей происходит на внутренней магистрали коммутатора. Пример сети рабочей группы такой структуры приведен на рис. 2.89.

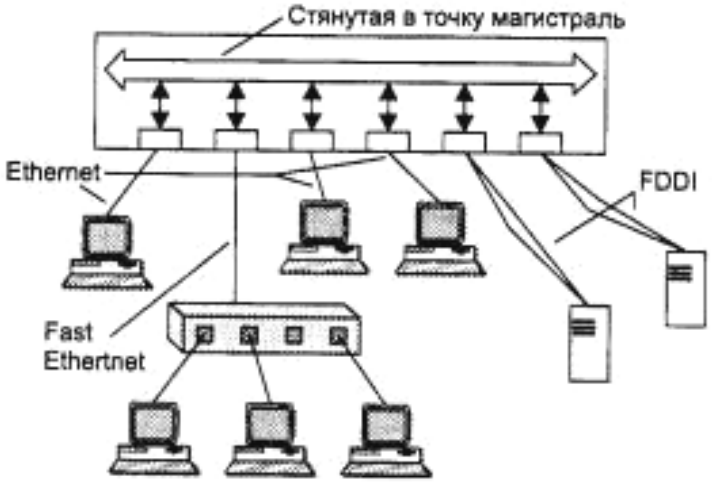


Рис. 2.89. Структура сети со стянутой в точку магистралью

Преимуществом такой структуры является высокая производительность магистрали. Так как для коммутатора производительность внутренней шины или схемы общей памяти, объединяющей модули портов, в несколько гигабит в секунду не является редкостью, то магистраль сети может быть весьма быстродействующей, причем ее скорость не зависит от применяемых в сети протоколов и может быть повышена с помощью замены одной модели коммутатора на другую.

Положительной чертой такой схемы является не только высокая скорость магистрали, но и ее протокольная независимость. На внутренней магистрали коммутатора в независимом формате одновременно могут передаваться данные различных протоколов, например Ethernet, FDDI и Fast Ethernet, как это изображено на рис. 2.89. Подключение нового узла с новым протоколом часто требует не замены коммутатора, а просто добавления соответствующего интерфейсного модуля, поддерживающего этот протокол.

Если к каждому порту коммутатора в такой схеме подключен только один узел, то такая схема будет соответствовать микросегментированной сети.

Распределенная магистраль на коммутаторах. В сетях больших зданий или кампусов структура с коллапсированной магистралью не всегда рациональна или возможна. Такая структура приводит к протяженным кабельным системам, связывающим конечные узлы или коммутаторы сетей рабочих групп с центральным коммутатором, шина которого и является магистралью сети. Высокая плотность кабелей и их высокая стоимость ограничивают применение стянутой в точку магистрали в таких сетях. Иногда, особенно в сетях кампусов, просто невозможно стянуть все кабели в одно помещение из-за ограничений на длину связей, накладываемых технологией (например, все реализации технологий локальных сетей на витой паре ограничивают протяженность кабелей в 100 м).

Поэтому в локальных сетях, покрывающих большие территории, часто используется другой вариант построения сети – с распределенной магистралью. Пример такой сети приведен на рис. 2.90.

Распределенная магистраль – это разделяемый сегмент сети, поддерживающий определенный протокол, к которому присоединяются коммутаторы сетей рабочих групп и отделов. На примере распределенная магистраль построена на основе двойного кольца FDDI, к которому подключены коммутаторы этажей. Коммутаторы этажей имеют большое количество портов Ethernet, трафик которых транслируется в трафик протокола FDDI, когда он передается по магистрали с этажа на этаж.

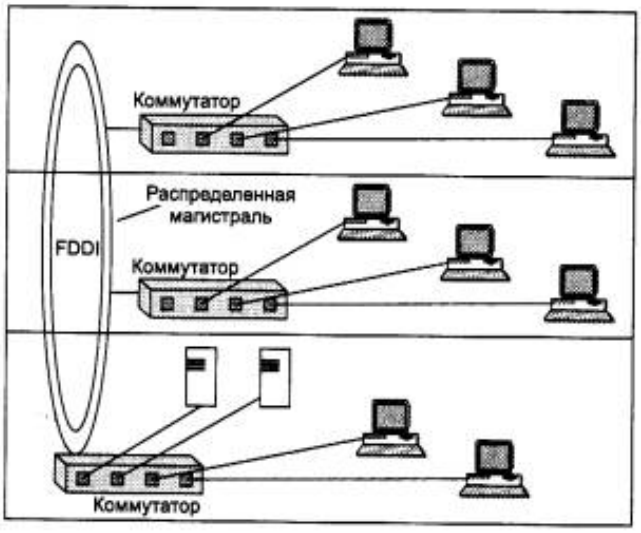


Рис. 2.90. Структура сети с распределенной магистралью

Распределенная магистраль упрощает связи между этажами, сокращает стоимость кабельной системы и преодолевает ограничения на расстояния.

Однако скорость магистрали в этом случае будет существенно ниже скорости магистрали на внутренней шине коммутатора. Причем скорость эта фиксированная и чаще всего не превышает 100 Мбит/с. Поэтому распределенная магистраль может применяться только при невысокой интенсивности трафика между этажами или зданиями. Широкое распространение в недалеком будущем технологии Gigabit Ethernet может снять это ограничение, что очень положительно скажется на структуре крупных сетей.

2.26. Принципы объединения сетей на основе протоколов сетевого уровня

В стандартной модели взаимодействия открытых систем в функции сетевого уровня входит решение следующих задач:

– передача пакетов между конечными узлами в составных сетях;

– выбор маршрута передачи пакетов, наилучшего по некоторому критерию;

– согласование разных протоколов канального уровня, использующихся в отдельных подсетях одной составной сети.

Протоколы сетевого уровня реализуются, как правило, в виде программных модулей и выполняются на конечных узлах – компьютерах, называемых хостами, а также на промежуточных узлах – маршрутизаторах,

называемых шлюзами. Функции маршрутизаторов могут выполнять как специализированные устройства, так и универсальные компьютеры с соответствующим программным обеспечением.

Ограничения мостов и коммутаторов

Создание сложной, структурированной сети, интегрирующей различные базовые технологии, может осуществляться и средствами канального уровня – для этого могут быть использованы некоторые типы мостов и коммутаторов. Мост или коммутатор разделяет сеть на сегменты, локализуя трафик внутри сегмента, что делает линии связи разделяемыми преимущественно между станциями данного сегмента. Тем самым сеть распадается на отдельные подсети, из которых могут быть построены составные сети достаточно крупных размеров.

Однако построение сложных сетей только на основе повторителей, мостов и коммутаторов имеет существенные ограничения и недостатки:

– в топологии получившейся сети должны отсутствовать петли. Действительно, мост/коммутатор может решать задачу доставки пакета адресату только тогда, когда между отправителем и получателем существует единственный путь. В то же время наличие избыточных связей, которые и образуют петли, часто необходимо для лучшей балансировки нагрузки, а также для повышения надежности сети за счет образования резервных путей;

– логические сегменты сети, расположенные между мостами или коммутаторами, слабо изолированы друг от друга, а именно не защищены от широковещательных штормов. Если какая-либо станция посылает широковещательное сообщение, то это сообщение передается всем станциям всех логических сегментов сети;

– в сетях, построенных на основе мостов и коммутаторов, достаточно сложно решается задача управления трафиком на основе значения данных, содержащихся в пакете. В таких сетях это возможно только с помощью пользовательских фильтров, для задания которых администратору приходится иметь дело с двоичным представлением содержимого пакетов;

– реализация транспортной подсистемы только средствами физического и канального уровней, к которым относятся мосты и коммутаторы, приводит к недостаточно гибкой, одноуровневой системе адресации – в качестве адреса назначения используется MAC-адрес, жестко связанный с сетевым адаптером;

– возможностью трансляции протоколов канального уровня обладают далеко не все типы мостов и коммутаторов, к тому же эти возможно-

сти ограничены. В частности, в объединяемых сетях должны совпадать максимально допустимые размеры полей данных в кадрах, т. к. мостами и коммутаторами не поддерживается функция фрагментации кадров.

Наличие серьезных ограничений у протоколов канального уровня показывает, что построение на основе средств этого уровня больших неоднородных сетей является весьма проблематичным. Естественное решение в этих случаях – это привлечение средств более высокого, сетевого уровня.

Понятие internetworking

Основная идея введения сетевого уровня состоит в следующем. Сеть в общем случае рассматривается как совокупность нескольких сетей и называется составной сетью или интерсетью (*internetwork* или *internet*). Сети, входящие в составную сеть, называются подсетями (*subnet*), составляющими сетями или просто сетями (рис. 2.91).

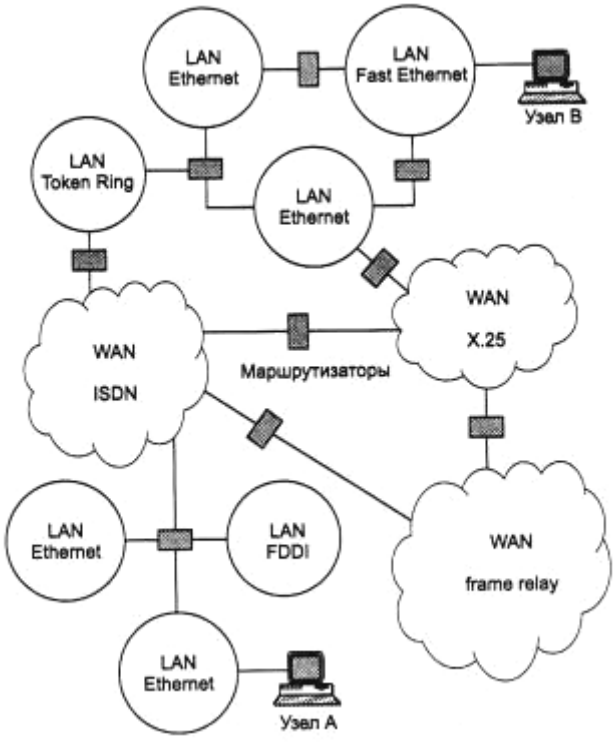


Рис. 2.91. Архитектура составной сети

Подсети соединяются между собой маршрутизаторами. Компонентами составной сети могут являться как локальные, так и глобальные сети. Внутренняя структура каждой сети на рисунке не показана, т. к. она не имеет значения при рассмотрении сетевого протокола. Все узлы в пределах

одной подсети взаимодействуют, используя единую для них технологию. Так, в составную сеть, показанную на рисунке, входят несколько сетей разных технологий: локальные сети Ethernet, Fast Ethernet, Token Ring, FDDI и глобальные сети frame relay, X.25, ISDN. Каждая из этих технологий достаточна для того, чтобы организовать взаимодействие всех узлов в своей подсети, но не способна построить информационную связь между произвольно выбранными узлами, принадлежащими разным подсетям, например, между узлом A и узлом B. Следовательно, для организации взаимодействия между любой произвольной парой узлов этой «большой» составной сети требуются дополнительные средства. Такие средства и предоставляет сетевой уровень.

Сетевой уровень выступает в качестве координатора, организующего работу всех подсетей, лежащих на пути продвижения пакета по составной сети. Для перемещения данных в пределах подсетей сетевой уровень обращается к используемым в этих подсетях технологиям.

Несмотря на то, что многие технологии локальных сетей (Ethernet, Token Ring, FDDI, Fast Ethernet и др.) используют одну и ту же систему адресации узлов на основе MAC-адресов, существует немало технологий (X.25, ATM, frame relay), в которых применяются другие схемы адресации. Адреса, присвоенные узлам в соответствии с технологиями подсетей, называют локальными. Чтобы сетевой уровень мог выполнить свою задачу, ему необходима собственная система адресации, не зависящая от способов адресации узлов в отдельных подсетях, которая позволила бы на сетевом уровне универсальными и однозначными способами идентифицировать любой узел составной сети.

Естественным способом формирования сетевого адреса является уникальная нумерация всех подсетей составной сети и нумерация всех узлов в пределах каждой подсети. Таким образом, сетевой адрес представляет собой пару: номер сети (подсети) и номер узла.

В качестве номера узла может выступать либо локальный адрес этого узла (такая схема принята в стеке IPX/SPX), либо некоторое число, никак не связанное с локальной технологией, которое однозначно идентифицирует узел в пределах данной подсети. В первом случае сетевой адрес становится зависимым от локальных технологий, что ограничивает его применение. Например, сетевые адреса IPX/SPX рассчитаны на работу в составных сетях, объединяющих сети, в которых используются только MAC-адреса или адреса аналогичного формата. Второй подход более универсален, он характерен для стека TCP/IP. И в том, и другом случае каждый узел составной сети имеет наряду со своим локальным адресом еще один – универсальный сетевой адрес.

Данные, которые поступают на сетевой уровень и которые необходимо передать через составную сеть, снабжаются заголовком сетевого уровня. Данные вместе с заголовком образуют пакет. Заголовок пакета сетевого уровня имеет унифицированный формат, не зависящий от форматов кадров канального уровня тех сетей, которые могут входить в объединенную сеть, и несет наряду с другой служебной информацией данные о номере сети, которой предназначается этот пакет. Сетевой уровень определяет маршрут и перемещает пакет между подсетями.

При передаче пакета из одной подсети в другую пакет сетевого уровня, инкапсулированный в прибывший канальный кадр первой подсети, освобождается от заголовков этого кадра и окружается заголовками кадра канального уровня следующей подсети. Информацией, на основе которой делается эта замена, являются служебные поля пакета сетевого уровня. В поле адреса назначения нового кадра указывается локальный адрес следующего маршрутизатора.

Основным полем заголовка сетевого уровня является номер сети-адресата. В рассмотренных ранее протоколах локальных сетей такого поля в кадрах предусмотрено не было – предполагалось, что все узлы принадлежат одной сети. Явная нумерация сетей позволяет протоколам сетевого уровня составлять точную карту межсетевых связей и выбирать рациональные маршруты при любой их топологии, в том числе альтернативные маршруты, если они имеются, что не умеют делать мосты и коммутаторы.

Кроме номера сети заголовок сетевого уровня должен содержать и другую информацию, необходимую для успешного перехода пакета из сети одного типа в сеть другого типа. К такой информации может относиться, например:

– номер фрагмента пакета, необходимый для успешного проведения операций сборки-разборки фрагментов при соединении сетей с разными максимальными размерами пакетов;

– время жизни пакета, указывающее, как долго он путешествует по интерсети, – это время может использоваться для уничтожения «заблудившихся» пакетов;

– качество услуги – критерий выбора маршрута при межсетевых передачах – например, узел-отправитель может потребовать передать пакет с максимальной надежностью, возможно, в ущерб времени доставки.

Когда две или более сети организуют совместную транспортную службу, то такой режим взаимодействия обычно называют межсетевым взаимодействием (*internetworking*).

Принципы маршрутизации

Важнейшей задачей сетевого уровня является маршрутизация – передача пакетов между двумя конечными узлами в составной сети.

Рассмотрим принципы маршрутизации на примере составной сети, изображенной на рис. 2.92.

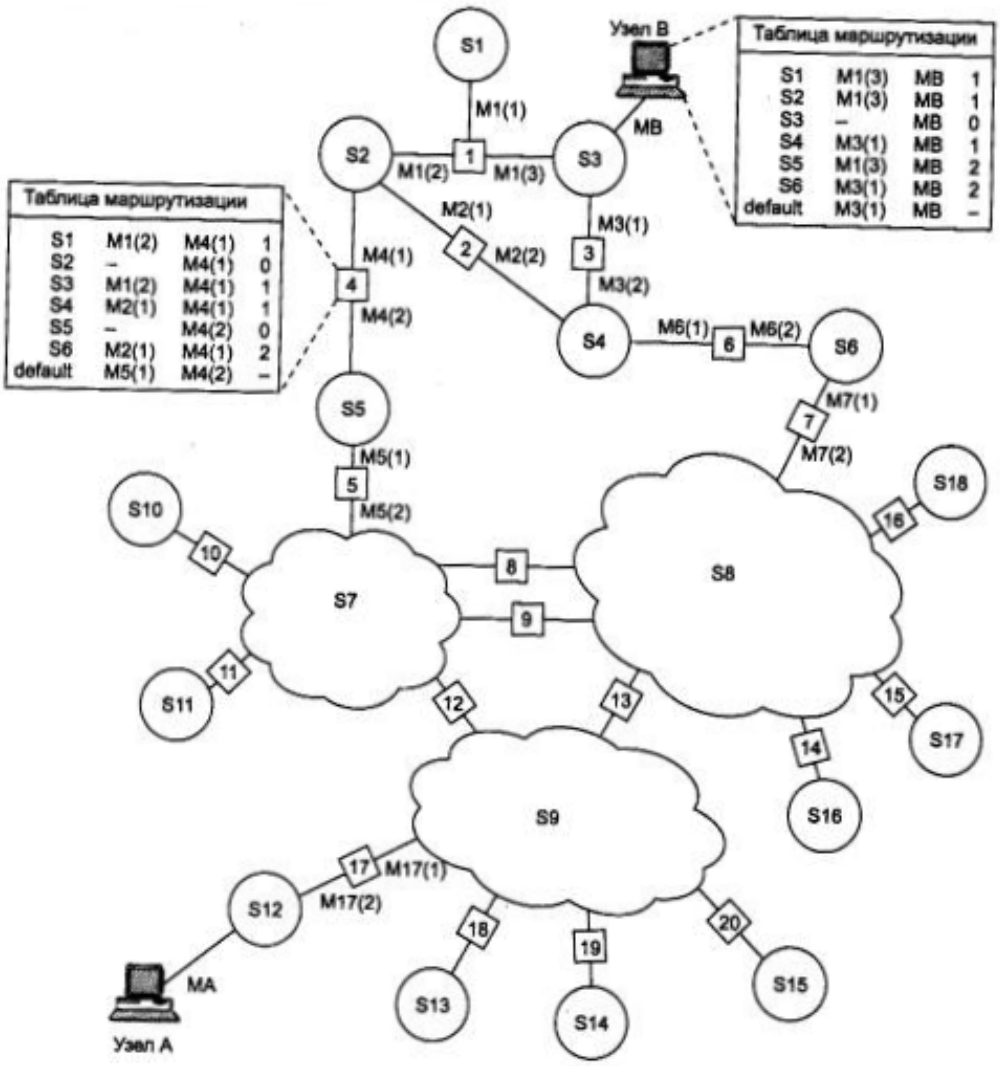


Рис. 2.92. Принципы маршрутизации в составной сети

В этой сети 20 маршрутизаторов объединяют 18 сетей в общую сеть, S1, S2, ... , S18 – это номера сетей. Маршрутизаторы имеют по нескольку портов (по крайней мере, по два), к которым присоединяются сети. Каждый порт маршрутизатора можно рассматривать как отдельный узел сети – он имеет собственный сетевой адрес и собственный локальный адрес в той подсети, которая к нему подключена. Например, маршрутизатор под номером 1 имеет три порта, к которым подключены сети S1, S2, S3. На рисунке сетевые адреса этих портов обозначены как M1(1), M1 (2) и M1(3). Порт

M1(1) имеет локальный адрес в сети с номером S1, порт M1 (2) – в сети S2, а порт M1(3) – в сети S3. Таким образом, маршрутизатор можно рассматривать как совокупность нескольких узлов, каждый из которых входит в свою сеть. Как единое устройство маршрутизатор не имеет ни отдельного сетевого адреса, ни какого-либо локального адреса.

В сложных составных сетях почти всегда существует несколько альтернативных маршрутов для передачи пакетов между двумя конечными узлами. Маршрут – это последовательность маршрутизаторов, которые должен пройти пакет от отправителя до пункта назначения. Так, пакет, отправленный из узла A в узел B, может пройти через маршрутизаторы 17, 12, 5, 4 и 1 или маршрутизаторы 17, 13, 7, 6 и 3. Также пристутствуют и другие маршрутоы между узлами A и B.

Задачу выбора маршрута из нескольких возможных решают маршрутизаторы, а также конечные узлы. Маршрут выбирается на основании имеющейся у этих устройств информации о текущей конфигурации сети, а также на основании указанного критерия выбора маршрута. Обычно в качестве критерия выступает задержка прохождения маршрута отдельным пакетом или средняя пропускная способность маршрута для последовательности пакетов. Часто также используется весьма простой критерий, учитывающий только количество пройденных в маршруте промежуточных маршрутизаторов (хопов).

Чтобы по адресу сети назначения можно было выбрать рациональный маршрут дальнейшего следования пакета, каждый конечный узел и маршрутизатор анализируют специальную информационную структуру, которая называется таблицей маршрутизации. Используя условные обозначения для сетевых адресов маршрутизаторов и номеров сетей в том виде, как они приведены на рис. 2.92, посмотрим, как могла бы выглядеть таблица маршрутизации, например, в маршрутизаторе 4 (табл. 2.4).

Таблица 2.4

Таблица маршрутизации маршрутизатора 4

Номер сети назначения	Сетевой адрес следующего маршрутизатора	Сетевой адрес выходного порта	Расстояние до сети назначения
S1	M1(2)	M4(1)	1
S2	–	M4(1)	0 (подсоединена)
S3	M1(2)	M4(1)	1
S4	M2(1)	M4(1)	1
S5		M4(2)	0 (подсоединена)
S6	M2(1)	M4(1)	2
Default	M5(2)	M4(2)	–

В первом столбце таблицы перечисляются номера сетей, входящих в интерсеть. В каждой строке таблицы следом за номером сети указывается сетевой адрес следующего маршрутизатора (более точно – сетевой адрес соответствующего порта следующего маршрутизатора), на который надо направить пакет, чтобы тот передвигался по направлению к сети с данным номером по рациональному маршруту.

Когда на маршрутизатор поступает новый пакет, номер сети назначения, извлеченный из поступившего кадра, последовательно сравнивается с номерами сетей из каждой строки таблицы. Строка с совпавшим номером сети указывает, на какой ближайший маршрутизатор следует направить пакет. Например, если на какой-либо порт маршрутизатора 4 поступает пакет, адресованный в сеть S6, то из таблицы маршрутизации следует, что адрес следующего маршрутизатора – M2(1), т. е. очередным этапом движения данного пакета будет движение к порту 1 маршрутизатора 2.

Поскольку пакет может быть адресован в любую сеть составной сети, может показаться, что каждая таблица маршрутизации должна иметь записи обо всех сетях, входящих в составную сеть. Но при таком подходе в случае крупной сети объем таблиц маршрутизации может оказаться очень большим, что повлияет на время их просмотра, потребует много места для хранения и т. п. Поэтому на практике число записей в таблице маршрутизации стараются уменьшить за счет использования специальной записи – *«маршрутизатор по умолчанию» (default)*. Действительно, если принять во внимание топологию составной сети, то в таблицах маршрутизаторов, находящихся на периферии составной сети, достаточно записать номера сетей, непосредственно подсоединенных к данному маршрутизатору или расположенных поблизости, на тупиковых маршрутах. Обо всех же остальных сетях можно сделать в таблице единственную запись, указывающую на маршрутизатор, через который пролегает путь ко всем этим сетям. Такой маршрутизатор называется маршрутизатором по умолчанию, а вместо номера сети в соответствующей строке помещается особая запись, например default. В нашем примере таким маршрутизатором по умолчанию для сети S5 является маршрутизатор 5, точнее, его порт M5(1). Это означает, что путь из сети S5 почти ко всем сетям большой составной сети пролегает через этот порт маршрутизатора.

Перед тем как передать пакет следующему маршрутизатору, текущий маршрутизатор должен определить, на какой из нескольких собственных портов он должен поместить данный пакет. Для этого служит третий столбец таблицы маршрутизации. Еще раз подчеркнем, что каждый порт идентифицируется собственным сетевым адресом.

Некоторые реализации сетевых протоколов допускают наличие в таблице маршрутизации сразу нескольких строк, соответствующих одному и тому же адресу сети назначения. В этом случае при выборе маршрута принимается во внимание столбец «Расстояние до сети назначения». При этом под расстоянием понимается любая метрика, используемая в соответствии с заданным в сетевом пакете критерием (часто называемым классом сервиса). Расстояние может измеряться хопами, временем прохождения пакета по линиям связи, какой-либо характеристикой надежности линий связи на данном маршруте или другой величиной, отражающей качество данного маршрута по отношению к заданному критерию. Если маршрутизатор поддерживает несколько классов сервиса пакетов, то таблица маршрутов составляется и применяется отдельно для каждого вида сервиса (критерия выбора маршрута).

В табл. 2.4 расстояние между сетями измерялось хопами. Расстояние для сетей, непосредственно подключенных к портам маршрутизатора, здесь принимается равным 0, однако в некоторых реализациях отсчет расстояний начинается с 1.

Наличие нескольких маршрутов к одному узлу делает возможным передачу трафика к этому узлу параллельно по нескольким каналам связи, это повышает пропускную способность и надежность сети.

Задачу маршрутизации решают не только промежуточные узлы – маршрутизаторы, но и конечные узлы – компьютеры. Средства сетевого уровня, установленные на конечном узле, при обработке пакета должны, прежде всего, определить, направляется ли он в другую сеть или адресован какому-нибудь узлу данной сети. Если номер сети назначения совпадает с номером данной сети, то для данного пакета не требуется решать задачу маршрутизации. Если же номера сетей отправления и назначения не совпадают, то маршрутизация нужна. Таблицы маршрутизации конечных узлов полностью аналогичны таблицам маршрутизации, хранящимся на маршрутизаторах.

Обратимся снова к сети, изображенной на рис. 2.92. Таблица маршрутизации для конечного узла B могла бы выглядеть следующим образом (табл. 2.5). Здесь MB – сетевой адрес порта компьютера B. На основании этой таблицы конечный узел B выбирает, на какой из двух имеющихся в локальной сети S3 маршрутизаторов следует посылать тот или иной пакет.

Конечные узлы в еще большей степени, чем маршрутизаторы, пользуются приемом маршрутизации по умолчанию. Хотя они также в общем случае имеют в своем распоряжении таблицу маршрутизации, ее объем обычно незначителен, что объясняется периферийным расположением всех конечных узлов.

Таблица 2.5

Таблица маршрутизации конечного узла B

Номер сети назначения	Сетевой адрес следующего маршрутизатора	Сетевой адрес выходного порта	Расстояние до сети назначения
S1	M1(3)	MB	1
S2	M1(3)	MB	1
S3	–	MB	0
S4	M3(1)	MB	1
S5	M1(3)	MB	2
S6	M3(1)	MB	2
Default	M3(1)	MB	–

Конечный узел часто вообще работает без таблицы маршрутизации, имея только сведения об адресе маршрутизатора по умолчанию. При наличии одного маршрутизатора в локальной сети этот вариант – единственно возможный для всех конечных узлов. Но даже при наличии нескольких маршрутизаторов в локальной сети, когда перед конечным узлом стоит проблема их выбора, задание маршрута по умолчанию часто используется в компьютерах для сокращения объема их таблицы маршрутизации.

2.27. Адресация в IP-сетях

Типы адресов стека TCP/IP

В стеке TCP/IP используются три типа адресов: локальные (называемые также аппаратными), IP-адреса и символьные доменные имена.

В терминологии TCP/IP под *локальным адресом* понимается такой тип адреса, который используется средствами базовой технологии для доставки данных в пределах подсети, являющейся элементом составной интерсети. В разных подсетях допустимы разные сетевые технологии, разные стеки протоколов, поэтому при создании стека TCP/IP предполагалось наличие разных типов локальных адресов. Если подсетью интерсети является локальная сеть, то локальный адрес – это MAC-адрес. MAC-адрес назначается сетевым адаптерам и сетевым интерфейсам маршрутизаторов. MAC-адреса назначаются производителями оборудования и являются уникальными, т. к. управляются централизованно. Для всех существующих технологий локальных сетей MAC-адрес имеет формат 6 байт, например 11-AO-17-3D-BC-01. Однако протокол IP может работать и над протоколами более высокого уровня, например над протоколом IPX или X.25. В этом слу-

чае локальными адресами для протокола IP, соответственно, будут адреса IPX и X.25. Следует учесть, что компьютер в локальной сети может иметь несколько локальных адресов даже при одном сетевом адаптере. Некоторые сетевые устройства не имеют локальных адресов. Например, к таким устройствам относятся глобальные порты маршрутизаторов, предназначенные для соединений типа «точка – точка».

IP-адреса представляют собой основной тип адресов, на основании которых сетевой уровень передает пакеты между сетями. Эти адреса состоят из 4 байт, например 109.26.17.100. IP-адрес назначается администратором во время конфигурирования компьютеров и маршрутизаторов. IP-адрес состоит из двух частей: номера сети и номера узла. Номер сети может быть выбран администратором произвольно либо назначен по рекомендации специального подразделения Internet (Internet Network Information Center, InterNIC), если сеть должна работать как составная часть Internet. Обычно поставщики услуг Internet получают диапазоны адресов у подразделений InterNIC, а затем распределяют их между своими абонентами. Номер узла в протоколе IP назначается независимо от локального адреса узла. Маршрутизатор по определению входит сразу в несколько сетей. Поэтому каждый порт маршрутизатора имеет собственный IP-адрес. Конечный узел также может входить в несколько IP-сетей. В этом случае компьютер должен иметь несколько IP-адресов, по числу сетевых связей. Таким образом, IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.

Символьные доменные имена. Символьные имена в IP-сетях называются доменными и строятся по иерархическому признаку. Составляющие полного символьного имени в IP-сетях разделяются точкой и перечисляются в следующем порядке: сначала простое имя конечного узла, затем имя группы узлов (например, имя организации), затем имя более крупной группы (поддомена) и так до имени домена самого высокого уровня (например, домена, объединяющего организации по географическому принципу: RU – Россия, UK – Великобритания, SU – США). Примером доменного имени может служить имя base2.sales.zil.ru. Между доменным именем и IP-адресом узла нет никакого алгоритмического соответствия, поэтому необходимо использовать какие-то дополнительные таблицы или службы, чтобы узел сети однозначно определялся как по доменному имени, так и по IP-адресу. В сетях TCP/IP используется специальная распределенная служба Domain Name System (DNS), которая устанавливает это соответствие на основании настраиваемых администраторами таблиц соответствия. Поэтому доменные имена называют также DNS-именами.

Классы IP-адресов

IP-адрес имеет длину 4 байта и обычно записывается в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками, например, 128.10.2.30 – традиционная десятичная форма представления адреса.

Адрес состоит из двух логических частей: номера сети и номера узла в сети. Какая часть адреса относится к номеру сети, а какая – к номеру узла, определяется значениями первых бит адреса. Значения этих бит являются также признаками того, к какому классу относится тот или иной IP-адрес.

На рис. 2.93 показана структура IP-адреса разных классов.

Рис. 2.93. Структура IP-адреса

Если адрес начинается с 0, то сеть относят к классу A и номер сети занимает один байт, остальные 3 байта интерпретируются как номер узла в сети. Сети класса A имеют номера в диапазоне от 1 до 126. (Номер 0 не используется, а номер 127 зарезервирован для специальных целей, о чем будет сказано ниже). Сетей класса A немного, зато количество узлов в них может достигать 224, т. е. 16 777 216 узлов.

Если первые два бита адреса равны 10, то сеть относится к классу B. В сетях класса B под номер сети и под номер узла отводится по 16 бит, т. е. по 2 байта. Таким образом, сеть класса B является сетью средних размеров с максимальным числом узлов 216, что составляет 65 536 узлов.

Если адрес начинается с последовательности 110, то это сеть класса C. В этом случае под номер сети отводится 24 бита, а под номер узла –

8 бит. Сети этого класса наиболее распространены, число узлов в них ограничено 28, то есть 256 узлами.

Если адрес начинается с последовательности 1110, то он является адресом класса D и обозначает особый, групповой адрес – *multicast*. Если в пакете в качестве адреса назначения указан адрес класса D, то такой пакет должны получить все узлы, которым присвоен данный адрес.

Если адрес начинается с последовательности 11110, то это значит, что данный адрес относится к классу E. Адреса этого класса зарезервированы для будущих применений.

В табл. 2.6 приведены диапазоны номеров сетей и максимальное число узлов, соответствующих каждому классу сетей.

Таблица 2.6

Характеристики адресов разного класса

Класс	Первые биты	Наименьший номер сети	Наибольший номер сети	Максимальное число узлов в сети
A	0	1.0.0.0	126.0.0.0	2^{24}
B	10	128.0.0.0	191.255.0.0	2^{16}
C	110	192.0.1.0	223.255.255.0	2^{8}
D	1110	224.0.0.0	239.255.255.255	Multicast
E	11110	240.0.0.0	247.255.255.255	Зарезервирован

Большие сети получают адреса класса A, средние – класса B, а маленькие – класса C.

Особые IP-адреса

В протоколе IP существует несколько соглашений об особой интерпретации IP-адресов:

– если весь IP-адрес состоит только из двоичных нулей, то он обозначает адрес того узла, который сгенерировал этот пакет; этот режим используется только в некоторых сообщениях ICMP;

– если в поле номера сети стоят только нули, то по умолчанию считается, что узел назначения принадлежит той же самой сети, что и узел, который отправил пакет;

– если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такая рассылка называется ограниченным широковещательным сообщением (*limited broadcast*);

– если в поле номера узла назначения стоят только единицы, то пакет, имеющий такой адрес, рассылается всем узлам сети с заданным номе-

ром сети. Например, пакет с адресом 192.190.21.255 доставляется всем узлам сети 192.190.21.0. Такая рассылка называется *широковещательным сообщением (broadcast).*

При адресации необходимо учитывать те ограничения, которые вносятся особым назначением некоторых IP-адресов. Так, ни номер сети, ни номер узла не может состоять только из одних двоичных единиц или только из одних двоичных нулей. Отсюда следует, что максимальное количество узлов, приведенное в таблице для сетей каждого класса, на практике должно быть уменьшено на 2. Например, в сетях класса С под номер узла отводится 8 бит, которые позволяют задавать 256 номеров: от 0 до 255. Однако на практике максимальное число узлов в сети класса С не может превышать 254, т. к. адреса 0 и 255 имеют специальное назначение. Из этих же соображений следует, что конечный узел не может иметь адрес типа 98.255.255.255, поскольку номер узла в этом адресе класса А состоит из одних двоичных единиц.

Особый смысл имеет IP-адрес, первый октет которого равен 127. Он используется для тестирования программ и взаимодействия процессов в пределах одной машины. Когда программа посылает данные по IP-адресу 127.0.0.1, то образуется «петля». Данные не передаются по сети, а возвращаются модулям верхнего уровня как только что принятые. Поэтому в IP-сети запрещается присваивать машинам IP-адреса, начинающиеся со 127. Этот адрес имеет название loopback. Можно отнести адрес 127.0.0.0 ко внутренней сети модуля маршрутизации узла, а адрес 127.0.0.1 – к адресу этого модуля на внутренней сети. На самом деле любой адрес сети 127.0.0.0 служит для обозначения своего модуля маршрутизации, а не только 127.0.0.1 (например, 127.0.0.3).

В протоколе IP нет понятия широковещательности в том смысле, в котором оно используется в протоколах канального уровня локальных сетей, когда данные должны быть доставлены абсолютно всем узлам. Как ограниченный широковещательный IP-адрес, так и широковещательный IP-адрес имеют пределы распространения в интерсети – они ограничены либо сетью, к которой принадлежит узел – источник пакета, либо сетью, номер которой указан в адресе назначения. Поэтому деление сети с помощью маршрутизаторов на части локализует широковещательный шторм пределами одной из составляющих общую сеть частей просто потому, что нет способа адресовать пакет одновременно всем узлам всех сетей составной сети.

Форма группового IP-адреса – *multicast* – означает, что данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу с номером, указанным в поле адреса. Узлы сами идентифицируют себя,

т. е. определяют, к какой из групп они относятся. Один и тот же узел может входить в несколько групп. Члены какой-либо группы *multicast* не обязательно должны принадлежать одной сети. В общем случае они могут распределяться по совершенно различным сетям, находящимся друг от друга на произвольном количестве хопов. Групповой адрес не делится на поля номера сети и узла и обрабатывается маршрутизатором особым образом.

Основное назначение multicast-адресов – распространение информации по схеме «один-ко-многим». Хост, который хочет передавать одну и ту же информацию многим абонентам, с помощью специального протокола IGMP (Internet Group Management Protocol) сообщает о создании в сети новой мультивещательной группы с определенным адресом. Машрутизаторы, поддерживающие мультивещательность, распространяют информацию о создании новой группы в сетях, подключенных к портам этого маршрутизатора. Хосты, которые хотят присоединиться к вновь создаваемой мультивещательной группе, сообщают об этом своим локальным маршрутизаторам, и те передают эту информацию хосту – инициатору создания новой группы.

Чтобы маршрутизаторы могли автоматически распространять пакеты с адресом multicast по составной сети, необходимо использовать в конечных маршрутизаторах модифицированные протоколы обмена маршрутной информацией, такие как, например, MOSPF (Multicast OSPF, аналог OSPF).

Групповая адресация предназначена для экономичного распространения в Internet или большой корпоративной сети аудио- или видеопрограмм, предназначенных сразу большой аудитории слушателей или зрителей.

Использование масок в IP-адресации

Традиционная схема деления IP-адреса на номер сети и номер узла основана на понятии класса, который определяется значениями нескольких первых бит адреса. Именно потому, что первый байт адреса 185.23.44.206 попадает в диапазон 128-191, мы можем сказать, что этот адрес относится к классу B, а значит, номером сети являются первые два байта, дополненные двумя нулевыми байтами, – 185.23.0.0, а номером узла – 0.0.44.206.

Однако возможно использование другого признака, с помощью которого можно более гибко устанавливать границу между номером сети и номером узла. В качестве такого признака широкое распространение получили маски. Маска – это число, которое используется в паре с IP-адресом; двоичная запись маски содержит единицы в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Поскольку номер сети яв-

ляется цельной частью адреса, единицы в маске также должны представлять непрерывную последовательность.

Для стандартных классов сетей маски имеют следующие значения:

- класс A – 11111111. 00000000. 00000000. 00000000 (255.0.0.0);
- класс B – 11111111. 11111111. 00000000. 00000000 (255.255.0.0);
- класс C – 11111111.11111111.11111111.00000000 (255.255.255.0).

Снабжая каждый IP-адрес маской, можно отказаться от понятий классов адресов и сделать более гибкой систему адресации. Например, если рассмотренный выше адрес 185.23.44.206 ассоциировать с маской 255.255.255.0, то номером сети будет 185.23.44.0, а не 185.23.0.0, как это определено системой классов.

В масках количество единиц в последовательности, определяющей границу номера сети, не обязательно должно быть кратным 8, чтобы повторять деление адреса на байты. Пусть, например, для IP-адреса 129.64.134.5 указана маска 255.255.128.0, т. е. в двоичном виде:

IP-адрес 129.64.134.5 – 10000001. 01000000.10000110. 00000101;

Маска 255.255.128.0 – 11111111.11111111.10000000. 00000000.

Если игнорировать маску, то в соответствии с системой классов адрес 129.64.134.5 относится к классу B, а значит, номером сети являются первые 2 байта – 129.64.0.0, а номером узла – 0.0.134.5.

Если же использовать для определения границы номера сети маску, то 17 последовательных единиц в маске, «наложенные» на IP-адрес, определяют в качестве номера сети в двоичном выражении число:

10000001. 01000000. 10000000. 00000000, или в десятичной форме записи – номер сети 129.64.128.0, а номер узла 0.0.6.5.

Механизм масок широко распространен в IP-маршрутизации, причем маски могут использоваться для самых разных целей. С их помощью администратор может структурировать свою сеть, не требуя от поставщика услуг дополнительных номеров сетей. На основе этого же механизма поставщики услуг могут объединять адресные пространства нескольких сетей путем введения так называемых «префиксов» с целью уменьшения объема таблиц маршрутизации и повышения за счет этого производительности маршрутизаторов.

Порядок распределения IP-адресов

Номера сетей назначаются либо централизованно, если сеть является частью Internet, либо произвольно, если сеть работает автономно. Номера узлов и в том, и в другом случае администратор может назначать по своему усмотрению, не выходя из разрешенного для этого класса сети диапазона.

Координирующую роль в централизованном распределении IP-адресов до некоторого времени играла организация InterNIC, однако с ростом сети задача распределения адресов стала слишком сложной и InterNIC делегировала часть своих функций другим организациям и крупным поставщикам услуг Internet.

Уже сравнительно давно наблюдается дефицит IP-адресов. Очень трудно получить адрес класса B, и практически невозможно стать обладателем адреса класса A. При этом надо отметить, что дефицит обусловлен не только ростом сетей, но и тем, что имеющееся множество IP-адресов используется нерационально. Очень часто владельцы сети класса C расходуют лишь небольшую часть из имеющихся у них 254 адресов. Рассмотрим пример, когда две сети необходимо соединить глобальной связью. В таких случаях в качестве канала связи используют два маршрутизатора, соединенных по схеме «точка – точка» (рис. 2.94). Для вырожденной сети, образованной каналом, связывающим порты двух смежных маршрутизаторов, приходится выделять отдельный номер сети, хотя в этой сети имеются всего 2 узла.

Рис. 2.94. Нерациональное использование пространства IP-адресов

Если же некоторая IP-сеть создана для работы в «автономном режиме», без связи с Internet, тогда администратор этой сети волен назначить ей произвольно выбранный номер. Но и в этой ситуации для того, чтобы избежать
каких-либо коллизий, в стандартах Internet определено несколько диапазонов адресов, рекомендуемых для локального использования. Эти адреса не обрабатываются маршрутизаторами Internet ни при каких условиях. Адреса, зарезервированные для локальных целей, выбраны из разных классов; в классе A это сеть 10.0.0.0, в классе B это диапазон из 16 номеров сетей 172.16.0.0 – 172.31.0.0, в классе C это диапазон из 255 сетей – 192.168.0.0 – 192.168.255.0.

Для смягчения проблемы дефицита адресов разработчики стека TCP/IP предлагают разные подходы. Принципиальным решением является

переход на новую версию IPv6, в которой резко расширяется адресное пространство за счет использования 16-байтных адресов. Однако и текущая версия IPv4 поддерживает некоторые технологии, направленные на более экономное расходование IP-адресов. Одной из таких технологий является технология масок и ее развитие – технология бесклассовой междоменной маршрутизации (Classless Inker – Domain Routing, CIDR). Технология CIDR отказывается от традиционной концепции разделения адресов протокола IP на классы, что позволяет получать в пользование столько адресов, сколько реально необходимо. Благодаря CIDR поставщик услуг получает возможность «нарезать» блоки из выделенного ему адресного пространства в точном соответствии с требованиями каждого клиента, при этом у него остается пространство для маневра на случай его будущего роста.

Другая технология, которая может быть использована для снятия дефицита адресов, это трансляция адресов (Network Address Translator, NAT). Узлам внутренней сети адреса назначаются произвольно (естественно, в соответствии с общими правилами, определенными в стандарте), так, как будто эта сеть работает автономно. Внутренняя сеть соединяется с Internet через некоторое промежуточное устройство (маршрутизатор, межсетевой экран). Это промежуточное устройство получает в свое распоряжение некоторое количество внешних «нормальных» IP-адресов, согласованных с поставщиком услуг или другой организацией, распределяющей IP-адреса. Промежуточное устройство способно преобразовывать внутренние адреса во внешние, используя для этого некие таблицы соответствия. Для внешних пользователей все многочисленные узлы внутренней сети выступают под несколькими внешними IP-адресами. При получении внешнего запроса это устройство анализирует его содержимое и при необходимости пересылает его во внутреннюю сеть, заменяя IP-адрес на внутренний адрес этого узла.

Автоматизация процесса назначения IP-адресов

Назначение IP-адресов узлам сети даже при не очень большом размере сети может представлять для администратора утомительную процедуру. Протокол Dynamic Host Configuration Protocol (DHCP) освобождает администратора от этих проблем, автоматизируя процесс назначения IP-адресов.

DHCP может поддерживать способ автоматического динамического распределения адресов, а также более простые способы ручного и автоматического статического назначения адресов. Протокол DHCP работает в соответствии с моделью клиент – сервер. Во время старта системы компь-

ютер, являющийся DHCP-клиентом, посылает в сеть широковещательный запрос на получение IP-адреса. DHCP-сервер откликается и посылает сообщение-ответ, содержащее IP-адрес. Предполагается, что DHCP-клиент и DHCP-сервер находятся в одной IP-сети.

При динамическом распределении адресов DHCP-сервер выдает адрес клиенту на ограниченное время, называемое временем аренды (*lease duration*), что дает возможность впоследствии повторно использовать этот IP-адрес для назначения другому компьютеру. Основное преимущество DHCP – автоматизация рутинной работы администратора по конфигурированию стека TCP/IP на каждом компьютере. Иногда динамическое разделение адресов позволяет строить IP-сеть, количество узлов в которой превышает количество имеющихся в распоряжении администратора IP-адресов.

В ручной процедуре назначения статических адресов активное участие принимает администратор, который предоставляет DHCP-серверу информацию о соответствии IP-адресов физическим адресам или другим идентификаторам клиентов. DHCP-сервер, пользуясь этой информацией, всегда выдает определенному клиенту назначенный администратором адрес.

При автоматическом статическом способе DHCP-сервер присваивает IP-адрес из пула наличных IP-адресов без вмешательства оператора. Границы пула назначаемых адресов задает администратор при конфигурировании DHCP-сервера. Адрес дается клиенту из пула в постоянное пользование, т. е. с неограниченным сроком аренды. Между идентификатором клиента и его IP-адресом по-прежнему, как и при ручном назначении, существует постоянное соответствие. Оно устанавливается в момент первого назначения DHCP-сервером IP-адреса клиенту. При всех последующих запросах сервер возвращает тот же самый IP-адрес.

DHCP обеспечивает надежный и простой способ конфигурации сети TCP/IP, гарантируя отсутствие дублирования адресов за счет централизованного управления их распределением. Администратор управляет процессом назначения адресов с помощью параметра «продолжительность аренды», который определяет, как долго компьютер может использовать назначенный IP-адрес, перед тем как снова запросить его от DHCP-сервера в аренду.

Примером работы протокола DHCP может служить ситуация, когда компьютер, являющийся DHCP-клиентом, удаляется из подсети. При этом назначенный ему IP-адрес автоматически освобождается. Когда компьютер подключается к другой подсети, то ему автоматически назначается но-

вый адрес. Ни пользователь, ни сетевой администратор не вмешиваются в этот процесс. Это свойство очень важно для мобильных пользователей.

DHCP-сервер может назначить клиенту не только IP-адрес клиента, но и другие параметры стека TCP/IP, необходимые для его эффективной работы, например, маску, IP-адрес маршрутизатора по умолчанию, IP-адрес сервера DNS, доменное имя компьютера и т. п.

Отображение IP-адресов на локальные адреса

Одной из главных задач, которая ставилась при создании протокола IP, было обеспечение совместной согласованной работы в сети, состоящей из подсетей, в общем случае использующих разные сетевые технологии. Непосредственно с решением этой задачи связан уровень межсетевых интерфейсов стека TCP/IP. На этом уровне определяются уже рассмотренные выше спецификации упаковки (инкапсуляции) IP-пакетов в кадры локальных технологий. Кроме этого, уровень межсетевых интерфейсов должен заниматься также крайне важной задачей отображения IP-адресов в локальные адреса.

Для определения локального адреса по IP-адресу используется протокол разрешения адреса (Address Resolution Protocol, ARP). Протокол ARP работает по-разному в зависимости от того, какой протокол канального уровня работает в данной сети – протокол локальной сети (Ethernet, Token Ring, FDDI) с возможностью широковещательного доступа одновременно ко всем узлам сети или же протокол глобальной сети (X.25, frame relay), как правило, не поддерживающий широковещательный доступ. Существует также протокол, решающий обратную задачу, – нахождение IP-адреса по известному локальному адресу. Он называется реверсивным ARP (Reverse Address Resolution Protocol, RARP) и используется при старте бездисковых станций, не знающих в начальный момент своего IP-адреса, но знающих адрес своего сетевого адаптера.

Необходимость в обращении к протоколу ARP возникает каждый раз, когда модуль IP передает пакет на уровень сетевых интерфейсов, например драйверу Ethernet. IP-адрес узла назначения известен модулю IP. Требуется на его основе найти MAC-адрес узла назначения.

Работа протокола ARP начинается с просмотра так называемой АКР-таблицы (табл. 2.7). Каждая строка таблицы устанавливает соответствие между IP-адресом и MAC-адресом. Для каждой сети, подключенной к сетевому адаптеру компьютера или к порту маршрутизатора, строится отдельная ARP-таблица.

Таблица 2.7

Пример ARP-таблицы

IP-адрес	MAC-адрес	Тип записи
194.85.135.75	008048EB7E60	Динамический
194.85.135.70	08005A21A722	Динамический
194.85.60.21	008048EB7567	Статический

Поле «Тип записи» может содержать одно из двух значений: «динамический» или «статический». Статические записи создаются вручную с помощью утилиты arp и не имеют срока устаревания, точнее, они существуют до тех пор, пока компьютер или маршрутизатор не будут выключены. Динамические же записи создаются модулем протокола ARP, использующим широковещательные возможности локальных сетевых технологий. Динамические записи должны периодически обновляться. Если запись не обновлялась в течение определенного времени (порядка нескольких минут), то она исключается из таблицы. Таким образом, в ARP-таблице содержатся записи не обо всех узлах сети, а только о тех, которые активно участвуют в сетевых операциях. Поскольку такой способ хранения информации называют кэшированием, ARP-таблицы иногда называют ARP-кэш.

После того как модуль IP обратился к модулю ARP с запросом на разрешение адреса, происходит поиск в ARP-таблице указанного в запросе IP-адреса. Если таковой адрес в ARP-таблице отсутствует, то исходящий IP-пакет, для которого нужно было определить локальный адрес, ставится в очередь. Далее протокол ARP формирует свой запрос (ARP-запрос), вкладывает его в кадр протокола канального уровня и рассылает запрос широковещательно.

Все узлы локальной сети получают ARP-запрос и сравнивают указанный там IP-адрес с собственным. В случае их совпадения узел формирует ARP-ответ, в котором указывает свой IP-адрес и свой локальный адрес, а затем отправляет его уже направленно, т. к. в ARP-запросе отправитель указывает свой локальный адрес. ARP-запросы и ответы используют один и тот же формат пакета.

Если в сети нет машины с искомым IP-адресом, то ARP-ответа не будет. Протокол IP уничтожает IP-пакеты, направляемые по этому адресу. (Заметим, что протоколы верхнего уровня не могут отличить случай повреждения сети Ethernet от случая отсутствия машины с искомым IP-адресом).

Если машина, сделавшая ARP-запрос, получила ответ, то модуль ARP анализирует ARP-ответ и добавляет запись в свою ARP-таблицу (табл. 2.8). В результате обмена двумя ARP-сообщениями модуль IP-узла

194.85.135.75 определил, что IP-адресу 194.85.135.65 соответствует MAC-адрес 00E0F77F1920. Новая запись в ARP-таблице появляется автоматически, спустя несколько миллисекунд после того, как она потребовалась.

Таблица 2.8

Обновленная ARP-таблица

IP-адрес	MAC-адрес	Тип записи
194.85.135.75	008048EB7E60	Динамический
194.85.135.70	08005A21A722	Динамический
194.85.60.21	008048EB7567	Статический
194.85.135.65	00E0F77F1920	Динамический

Отображение доменных имен на IP-адреса

Для идентификации компьютеров аппаратное и программное обеспечение в сетях TCP/IP полагается на IP-адреса, поэтому для доступа к сетевому ресурсу в параметрах программы вполне достаточно указать IP-адрес, чтобы программа правильно поняла, к какому хосту ей нужно обратиться. Например, команда ftp://192.45.66.17 будет устанавливать сеанс связи с нужным ftp-сервером, а команда http://203.23.106.33 откроет начальную страницу на корпоративном Web-сервере. Однако пользователи обычно предпочитают работать с символьными именами компьютеров, и операционные системы локальных сетей приучили их к этому удобному способу. Следовательно, в сетях TCP/IP должны существовать символьные имена хостов и механизм для установления соответствия между символьными именами и IP-адресами.

В операционных системах, которые первоначально разрабатывались для работы в локальных сетях, таких как Novell NetWare, Microsoft Windows или IBM OS/2, пользователи всегда работали с символьными именами компьютеров. Так как локальные сети состояли из небольшого числа компьютеров, то использовались так называемые плоские имена, состоящие из последовательности символов, не разделенных на части. Примерами таких имен являются NW1_1, mail2, MOSCOW_SALES_2. Для установления соответствия между символьными именами и MAC-адресами в этих операционных системах применялся механизм широковещательных запросов, подобный механизму запросов протокола ARP. Так, широковещательный способ разрешения имен реализован в протоколе NetBIOS, на котором были построены многие локальные ОС. Так называемые NetBIOS-имена стали на долгие годы одним из основных типов плоских имен в локальных сетях.

Для стека TCP/IP, рассчитанного в общем случае на работу в больших территориально распределенных сетях, подобный подход оказывается неэффективным по нескольким причинам.

Плоские имена не дают возможности разработать единый алгоритм обеспечения уникальности имен в пределах большой сети. В небольших сетях уникальность имен компьютеров обеспечивает администратор сети, записывая несколько десятков имен в журнале или файле. При росте сети задачу решают уже несколько администраторов, согласовывая имена между собой неформальным способом. Однако если сеть расположена в разных городах или странах, то администраторам каждой части сети нужно придумать способ именования, который позволил бы им давать имена новым компьютерам независимо от других администраторов, обеспечивая в то же время уникальность имен для всей сети. Самый надежный способ решения этой задачи – отказ от плоских имен в принципе.

Широковещательный способ установления соответствия между символьными именами и локальными адресами хорошо работает только в небольшой локальной сети, не разделенной на подсети. В крупных сетях, где общая широковещательность не поддерживается, нужен другой способ разрешения символьных имен. Обычно хорошей альтернативой широковещательности является применение централизованной службы, поддерживающей соответствие между различными типами адресов всех компьютеров сети. Компания Microsoft для своей корпоративной операционной системы Windows NT разработала централизованную службу WINS, которая поддерживает базу данных NetBIOS-имен и соответствующих им IP-адресов.

Для эффективной организации именования компьютеров в больших сетях естественным является применение иерархических составных имен.

В стеке TCP/IP применяется доменная система имен, которая имеет иерархическую древовидную структуру, допускающую использование в имени произвольного количества составных частей (рис. 2.95).

Иерархия доменных имен аналогична иерархии имен файлов, принятой во многих популярных файловых системах. Дерево имен начинается с корня, обозначаемого здесь точкой (.). Затем следует старшая символьная часть имени, вторая по старшинству символьная часть имени и т. д. Младшая часть имени соответствует конечному узлу сети. В отличие от имен файлов, при записи которых сначала указывается самая старшая составляющая, затем составляющая более низкого уровня и т. д., запись доменного имени начинается с самой младшей составляющей, а заканчивается

самой старшей. Составные части доменного имени отделяются друг от друга точкой. Например, в имени partnering.microsoft.com составляющая partnering является именем одного из компьютеров в домене Microsoft.com.

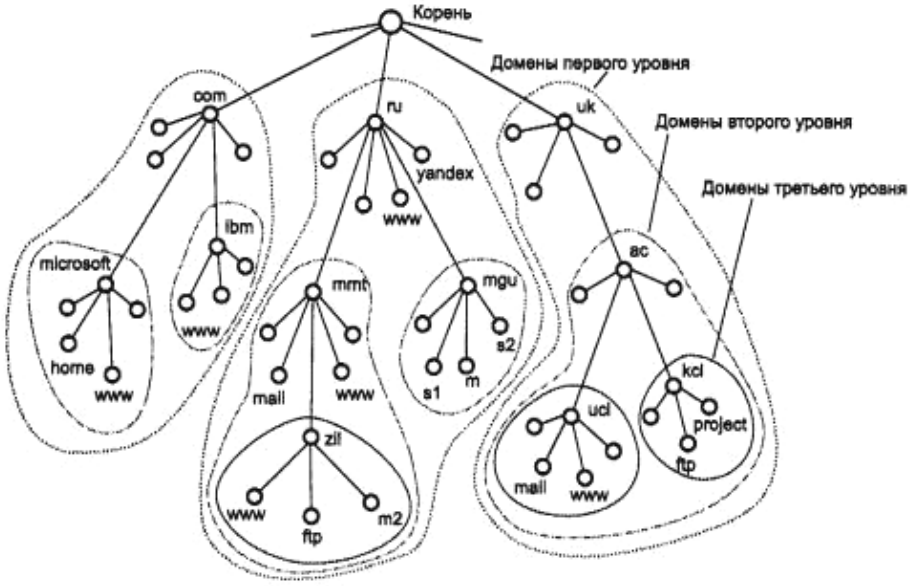


Рис. 2.95. Пространство доменных имен

Разделение имени на части позволяет разделить административную ответственность за назначение уникальных имен между различными людьми или организациями в пределах своего уровня иерархии. Разделение административной ответственности позволяет решить проблему образования уникальных имен без взаимных консультаций между организациями, отвечающими за имена одного уровня иерархии. Очевидно, что должна существовать одна организация, отвечающая за назначение имен верхнего уровня иерархии.

Совокупность имен, у которых несколько старших составных частей совпадают, образуют *домен имен (domain)*. Например, имена wwwl.zil.mmt.ru, ftp.zil.mmt.ru, yandex.ru и sl.mgu.ru входят в домен ru, т. к. все эти имена имеют одну общую старшую часть – имя ru. Другим примером является домен mgu.ru. Из представленных на рис. 2.95 имен в него входят имена sl.mgu.ru, s2.mgu.ru и rn.mgu.ru. Этот домен образуют имена, у которых две старшие части всегда равны rngu.ru. Имя www.mmt.ru в домен mgu.ru не входит, т. к. имеет отличающуюся составляющую mmt.

Если один домен входит в другой домен как его составная часть, то такой домен могут называть *поддоменом (subdomain)*, хотя название *домен* за ним также остается. Обычно поддомен называют по имени той его старшей составляющей, которая отличает его от других поддоменов. Например, поддомен mmt.ru обычно называют поддоменом (или доменом)

mmt. Имя поддомену назначает администратор вышестоящего домена. Хорошей аналогией домена является каталог файловой системы.

Если в каждом домене и поддомене обеспечивается уникальность имен следующего уровня иерархии, то и вся система имен будет состоять из уникальных имен.

По аналогии с файловой системой в доменной системе имен различают краткие имена, относительные имена и полные доменные имена. Краткое имя – это имя конечного узла сети – хоста или порта маршрутизатора. Краткое имя – это лист дерева имен. Относительное имя – это составное имя, начинающееся с некоторого уровня иерархии, но не самого верхнего. Например, wwwi.zil – это относительное имя. *Полное доменное имя (fully qualified domain name, FQJDN)* включает составляющие всех уровней иерархии, начиная от краткого имени и кончая корневой точкой: wwwl.zil.mmt.ru.

Необходимо подчеркнуть, что компьютеры входят в домен в соответствии со своими составными именами, при этом они могут иметь совершенно различные IP-адреса, принадлежащие к различным сетям и подсетям. Например, в домен mgu.ru могут входить хосты с адресами 132.13.34.15, 201.22.100.33,14.0.0.6. Доменная система имен реализована в сети Internet, но она может работать и как автономная система имен в крупной корпоративной сети, использующей стек TCP/IP, но не связанной с Internet.

В Internet корневой домен управляется центром InterNIC. Домены верхнего уровня назначаются для каждой страны, а также на организационной основе. Имена этих доменов должны следовать международному стандарту ISO 3166. Для обозначения стран используются трехбуквенные и двухбуквенные аббревиатуры, а для различных типов организаций – следующие обозначения:

- com – коммерческие организации (например, microsoft.com);
- edu – образовательные (например, mitedu);
- gov – правительственные организации (например, nsf.gov);
- org – некоммерческие организации (например, fidonet.org);
- net – организации, поддерживающие сети (например, nsf.net).

Каждый домен администрируется отдельной организацией, которая обычно разбивает свой домен на поддомены и передает функции администрирования этих поддоменов другим организациям. Чтобы получить доменное имя, необходимо зарегистрироваться в какой-либо организации, которой InterNIC делегировал свои полномочия по распределению имен доменов. В России такой организацией является РосНИИРОС, которая отвечает за делегирование имен поддоменов в домене ru.

Система доменных имен DNS

Соответствие между доменными именами и IP-адресами может устанавливаться как средствами локального хоста, так и средствами централизованной службы. На раннем этапе развития Internet на каждом хосте вручную создавался текстовый файл с известным именем hosts. Этот файл состоял из некоторого количества строк, каждая из которых содержала одну пару «IP-адрес – доменное имя», например 102.54.94.97 – rhino.acme.com.

По мере роста Internet файлы hosts также росли и создание масштабируемого решения для разрешения имен стало необходимостью.

Таким решением стала специальная служба – *система доменных имен (Domain Name System, DNS)*. DNS – это централизованная служба, основанная на распределенной базе отображений «доменное имя – IP-адрес». Служба DNS использует в своей работе протокол типа «клиент – сервер». В нем определены DNS-серверы и DNS-клиенты. DNS-серверы поддерживают распределенную базу отображений, а DNS-клиенты обращаются к серверам с запросами о разрешении доменного имени в IP-адрес.

Служба DNS использует текстовые файлы почти такого формата, как и файл hosts, и эти файлы администратор также подготавливает вручную. Однако служба DNS опирается на иерархию доменов, и каждый сервер службы DNS хранит только часть имен сети, а не все имена, как это происходит при использовании файлов hosts. При росте количества узлов в сети проблема масштабирования решается созданием новых доменов и поддоменов имен и добавлением в службу DNS новых серверов.

Для каждого домена имен создается свой DNS-сервер. Этот сервер может хранить отображения «доменное имя – IP-адрес» для всего домена, включая все его поддомены. Однако при этом решение оказывается плохо масштабируемым, т. к. при добавлении новых поддоменов нагрузка на этот сервер может превысить его возможности. Чаще сервер домена хранит только имена, которые заканчиваются на следующем ниже уровне иерархии по сравнению с именем домена. (Аналогично каталогу файловой системы, который содержит записи о файлах и подкаталогах, непосредственно в него «входящих».) Именно при такой организации службы DNS нагрузка по разрешению имен распределяется более-менее равномерно между всеми DNS-серверами сети. Например, в первом случае DNS-сервер домена mmtru будет хранить отображения для всех имен, заканчивающихся на mmt.ru: wwwl.zil.mmt.ru, ftp.zil.mmt.ru, mail.mmt.ru и т. д. Во втором случае этот сервер хранит отображения только имен типа mail.mmt.ru, www.mmt.ru, а все остальные отображения должны храниться на DNS-сервере поддомена zil.

Каждый DNS-сервер кроме таблицы отображений имен содержит ссылки на DNS-серверы своих поддоменов. Эти ссылки связывают отдельные DNS-серверы в единую службу DNS. Ссылки представляют собой IP-адреса соответствующих серверов. Для обслуживания корневого домена выделено несколько дублирующих друг друга DNS-серверов, IP-адреса которых являются широко известными (их можно узнать, например, в InterNIC).

Процедура разрешения DNS-имени во многом аналогична процедуре поиска файловой системой адреса файла по его символьному имени. Действительно, в обоих случаях составное имя отражает иерархическую структуру организации соответствующих справочников – каталогов файлов или таблиц DNS. Здесь домен и доменный DNS-сервер являются аналогом каталога файловой системы. Для доменных имен, так же как и для символьных имен файлов, характерна независимость именования от физического местоположения.

Процедура поиска адреса файла по символьному имени заключается в последовательном просмотре каталогов, начиная с корневого. При этом предварительно проверяется кэш и текущий каталог. Для определения IP-адреса по доменному имени также необходимо просмотреть все DNS-серверы, обслуживающие цепочку поддоменов, входящих в имя хоста, начиная с корневого домена. Существенным же отличием является то, что файловая система расположена на одном компьютере, а служба DNS по своей природе является распределенной.

Существуют две основные схемы разрешения DNS-имен. В первом варианте работу по поиску IP-адреса координирует DNS-клиент:

– DNS-клиент обращается к корневому DNS-серверу с указанием полного доменного имени;

– DNS-сервер отвечает, указывая адрес следующего DNS-сервера, обслуживающего домен верхнего уровня, заданный в старшей части запрошенного имени;

– DNS-клиент делает запрос следующего DNS-сервера, который отсылает его к DNS-серверу нужного поддомена, и т. д., пока не будет найден DNS-сервер, в котором имеется соответствие запрошенного имени IP-адресу. Этот сервер дает окончательный ответ клиенту.

Такая схема взаимодействия называется нерекурсивной или итеративной, когда клиент сам итеративно выполняет последовательность запросов к разным серверам имен. Так как эта схема загружает клиента достаточно сложной работой, то она применяется редко.

Во втором варианте реализуется рекурсивная процедура:

– DNS-клиент запрашивает локальный DNS-сервер, т. е. тот сервер, который обслуживает поддомен, к которому принадлежит имя клиента;

– если локальный DNS-сервер знает ответ, то он сразу же возвращает его клиенту – это может соответствовать случаю, когда запрошенное имя входит в тот же поддомен, что и имя клиента, а также может соответствовать случаю, когда сервер уже узнавал данное соответствие для другого клиента и сохранил его в своем кэше;

– если же локальный сервер не знает ответ, то он выполняет итеративные запросы к корневому серверу и т. д. точно так же, как это делал клиент в первом варианте; получив ответ, он передает его клиенту, который все это время просто ждал его от своего локального DNS-сервера.

В этой схеме клиент перепоручает работу своему серверу, поэтому схема называется косвенной или рекурсивной. Практически все DNS-клиенты используют рекурсивную процедуру.

Для ускорения поиска IP-адресов DNS-серверы широко применяют процедуру кэширования проходящих через них ответов. Чтобы служба DNS могла оперативно отрабатывать изменения, происходящие в сети, ответы кэшируются на определенное время – обычно от нескольких часов до нескольких дней.

2.28. Протокол IP

Основные функции протокола IP

Основу транспортных средств стека протоколов TCP/IP составляет протокол межсетевого взаимодействия (Internet Protocol, IP). Он обеспечивает передачу дейтаграмм от отправителя к получателям через объединенную систему компьютерных сетей.

Название этого протокола – Intrenet Protocol – отражает его суть: он должен передавать пакеты *между сетями*. В каждой очередной сети, лежащей на пути перемещения пакета, протокол IP вызывает средства транспортировки, принятые в этой сети, чтобы с их помощью передать этот пакет на маршрутизатор, ведущий к следующей сети, или непосредственно на узел-получатель.

Протокол IP относится к протоколам без установления соединений. Перед IP не ставится задача надежной доставки сообщений от отправителя к получателю. Протокол IP обрабатывает каждый IP-пакет как независимую единицу, не имеющую связи ни с какими другими IP-пакетами.

В протоколе IP нет механизмов, обычно применяемых для увеличения достоверности конечных данных: отсутствует квитирование – обмен подтверждениями между отправителем и получателем, нет процедуры упорядочивания, повторных передач или других подобных функций. Если во время продвижения пакета произошла какая-либо ошибка, то протокол IP по своей инициативе ничего не предпринимает для исправления этой ошибки. Например, если на промежуточном маршрутизаторе пакет был отброшен по причине истечения времени жизни или из-за ошибки в контрольной сумме, то модуль IP не пытается заново послать испорченный или потерянный пакет. Все вопросы обеспечения надежности доставки данных по составной сети в стеке TCP/IP решает протокол TCP, работающий непосредственно над протоколом IP. Именно TCP организует повторную передачу пакетов, когда в этом возникает необходимость.

Важной особенностью протокола IP, отличающей его от других сетевых протоколов (например, от сетевого протокола IPX), является его способность выполнять динамическую фрагментацию пакетов при передаче их между сетями с различными, максимально допустимыми значениями поля данных кадров MTU. Свойство фрагментации во многом способствовало тому, что протокол IP смог занять доминирующие позиции в сложных составных сетях.

Имеется прямая связь между функциональной сложностью протокола и сложностью заголовка пакетов, которые этот протокол использует. Это объясняется тем, что основные служебные данные, на основании которых протокол выполняет то или иное действие, переносятся между двумя модулями, реализующими этот протокол на разных машинах, именно в полях заголовков пакетов. Поэтому полезно изучить назначение каждого поля заголовка IP-пакета, и это изучение не только дает формальные знания о структуре пакета, но и объясняет все основные режимы работы протокола по обработке и передаче IP-дейтаграмм.

Структура IP-пакета

IP-пакет состоит из заголовка и поля данных. Заголовок, как правило, имеющий длину 20 байт, имеет следующую структуру (рис. 2.96).

Поле *Номер версии (Version)*, занимающее 4 бит, указывает версию протокола IP. Сейчас повсеместно используется версия 4 (IPv4) и готовится переход на версию 6 (IPv6).

Поле *Длина заголовка (IHL)* IP-пакета занимает 4 бит и указывает значение длины заголовка, измеренное в 32-битовых словах. Обычно заго-

ловок имеет длину в 20 байт (пять 32-битовых слов), но при увеличении объема служебной информации эта длина может быть увеличена за счет использования дополнительных байт в поле *Опции (IP Options)*. Наибольший заголовок занимает 60 октетов.

Рис. 2.96. Структура заголовка IP-пакета

Поле *Тип сервиса (Type of Service)* занимает один байт и задает приоритетность пакета и вид критерия выбора маршрута. Первые три бита этого поля образуют подполе *приоритета пакета (Precedence)*. Приоритет может иметь значения от самого низкого – 0 (нормальный пакет) до самого высокого – 7 (пакет управляющей информации). Маршрутизаторы и компьютеры могут принимать во внимание приоритет пакета и обрабатывать более важные пакеты в первую очередь. Поле *Тип сервиса* содержит также три бита, определяющие критерий выбора маршрута. Реально выбор осуществляется между тремя альтернативами: малой задержкой, высокой достоверностью и высокой пропускной способностью. Установленный бит D (*delay*) говорит о том, что маршрут должен выбираться для минимизации задержки доставки данного пакета, бит T – для максимизации пропускной способности, а бит R – для максимизации надежности доставки. Во многих сетях улучшение одного из этих параметров связано с ухудшением другого, кроме того, обработка каждого из них требует дополнительных вычислительных затрат. Поэтому редко когда имеет смысл устанавливать одновременно хотя бы два из этих трех критериев выбора маршрута. Зарезервированные биты имеют нулевое значение.

Поле *Общая длина (Total Length)* занимает 2 байта и означает общую длину пакета с учетом заголовка и поля данных. Максимальная длина пакета ограничена разрядностью поля, определяющего эту величину, и составляет 65 535 байт, однако в большинстве хост-компьютеров и сетей столь большие пакеты не используются. При передаче по сетям различного типа длина пакета выбирается с учетом максимальной длины пакета протокола нижнего уровня, несущего IP-пакеты. Если это кадры Ethernet, то выбираются пакеты с максимальной длиной в 1 500 байт, умещающиеся в поле данных кадра Ethernet. В стандарте предусматривается, что все хосты должны быть готовы принимать пакеты вплоть до 576 байт длиной (приходят ли они целиком или по фрагментам). Хостам рекомендуется отправлять пакеты размером более чем 576 байт, только если они уверены, что принимающий хост или промежуточная сеть готовы обслуживать пакеты такого размера.

Поле *Идентификатор пакета (Identification)* занимает 2 байта и используется для распознавания пакетов, образовавшихся путем фрагментации исходного пакета. Все фрагменты должны иметь одинаковое значение этого поля.

Поле *Флаги (Flags)* занимает 3 бита и содержит признаки, связанные с фрагментацией. Установленный бит *DF (Do not Fragment)* запрещает маршрутизатору фрагментировать данный пакет, а установленный бит *MF (More Fragments)* говорит о том, что данный пакет является промежуточным (не последним) фрагментом. Оставшийся бит зарезервирован.

Поле *Смещение фрагмента (Fragment Offset)* занимает 13 бит и задает смещение в байтах поля данных этого пакета от начала общего поля данных исходного пакета, подвергнутого фрагментации. Используется при сборке/разборке фрагментов пакетов при передаче их между сетями с различными величинами MTU. Смещение должно быть кратно 8 байт.

Поле *Время жизни (Time to Live)* занимает один байт и означает предельный срок, в течение которого пакет может перемещаться по сети. Время жизни данного пакета измеряется в секундах и задается источником передачи. На маршрутизаторах и в других узлах сети по истечении каждой секунды из текущего времени жизни вычитается единица; единица вычитается и в том случае, когда время задержки меньше секунды. Поскольку современные маршрутизаторы редко обрабатывают пакет дольше, чем за одну секунду, то время жизни можно считать равным максимальному числу узлов, которые разрешено пройти данному пакету до того, как он достигнет места назначения. Если параметр времени жизни станет нулевым до

того, как пакет достигнет получателя, этот пакет будет уничтожен. Время жизни можно рассматривать как часовой механизм самоуничтожения. Значение этого поля изменяется при обработке заголовка IP-пакета.

Идентификатор *Протокол верхнего уровня (Protocol)* занимает один байт и указывает, какому протоколу верхнего уровня принадлежит информация, размещенная в поле данных пакета (например, это могут быть сегменты протокола TCP, дейтаграммы UDP, пакеты ICMP или OSPF). Значения идентификаторов для различных протоколов приводятся в документе RFC «Assigned Numbers».

Контрольная сумма (Header Checksum) занимает 2 байта и рассчитывается только по заголовку. Поскольку некоторые поля заголовка меняют свое значение в процессе передачи пакета по сети (например, время жизни), контрольная сумма проверяется и повторно рассчитывается при каждой обработке IP-заголовка. Контрольная сумма – 16 бит – подсчитывается как дополнение к сумме всех 16-битовых слов заголовка. При вычислении контрольной суммы значение самого поля «контрольная сумма» устанавливается в нуль. Если контрольная сумма неверна, то пакет будет отброшен, как только ошибка будет обнаружена.

Поля *IP-адрес источника (Source IP Address)* и *IP-адрес назначения (Destination IP Address)* имеют одинаковую длину – 32 бита – и одинаковую структуру.

Поле *Опции (IP Options)* является необязательным и используется обычно только при отладке сети. Механизм опций предоставляет функции управления, которые необходимы или просто полезны при определенных ситуациях, однако он не нужен при обычных коммуникациях. Это поле состоит из нескольких подполей, каждое из которых может быть одного из восьми предопределенных типов. В этих подполях можно указывать точный маршрут прохождения маршрутизаторов, регистрировать проходимые пакетом маршрутизаторы, помещать данные системы безопасности, а также временные отметки. Так как число подполей может быть произвольным, то в конце поля *Опции* должно быть добавлено несколько байт для выравнивания заголовка пакета по 32-битной границе.

Поле *Выравнивание (Padding)* используется для того, чтобы убедиться в том, что IP-заголовок заканчивается на 32-битной границе. Выравнивание осуществляется нулями.

Ниже приведена распечатка значений полей заголовка одного из реальных IP-пакетов, захваченных в сети Ethernet средствами анализатора протоколов Microsoft Network Monitor.

IP Version = 4 (0x4)
IP Header Length = 20 (0x14)
IP Service Type = 0 (0x0)
IP Precedence = Routine
IP ...0.... = Normal Delay
IP ....0... = Normal Throughput
IP .....0.. = Normal Reliability
IP Total Length = 54 (0x36)
IP Identification = 31746 (0x7C02)
IP Flags Summary = 2 (0x2)
IP ..... 0 = Last fragment in datagram
IP ..... 1. = Cannot fragment datagram
IP Fragment Offset = 0 (0x0) bytes
IP Time to Live = 128 (0x80)
IP Protocol = TCP-Transmission Control
IP Checksum =0xEB86
IP Source Address = 194.85.135.75
IP Destination Address = 194.85.135.66
IP Data: Number of data bytes remaining = 34 (0x0022)

2.29. Протоколы маршрутизации в IP-сетях

Внутренние и внешние протоколы маршрутизации Internet

Internet изначально строилась как сеть, объединяющая большое количество существующих систем. С самого начала в ее структуре выделяли *магистральную сеть (core backbone network)*, а сети, присоединенные к магистрали, рассматривались как *автономные системы (autonomous systems, AS)*. Магистральная сеть и каждая из автономных систем имели свое собственное административное управление и собственные протоколы маршрутизации. Необходимо подчеркнуть, что автономная система и домен имен Internet – это разные понятия, которые служат разным целям. Автономная система объединяет сети, в которых под общим административным руководством одной организации осуществляется маршрутизация, а домен объединяет компьютеры (возможно, принадлежащие разным сетям), в которых под общим административным руководством одной организации осуществляется назначение уникальных символьных имен. Естественно,

области действия автономной системы и домена имен могут в частном случае совпадать, если одна организация выполняет обе указанные функции.

Общая схема архитектуры сети Internet показана на рис. 2.97. Далее маршрутизаторы мы будем называть шлюзами.

Шлюзы, которые используются для образования сетей и подсетей внутри автономной системы, называются *внутренними шлюзами (interior gateways)*, а шлюзы, с помощью которых автономные системы присоединяются к магистрали сети, называются *внешними шлюзами (exterior gateways)*. Магистраль сети также является автономной системой. Все автономные системы имеют уникальный 16-разрядный номер, который выделяется организацией, учредившей новую автономную систему – InterNIC.

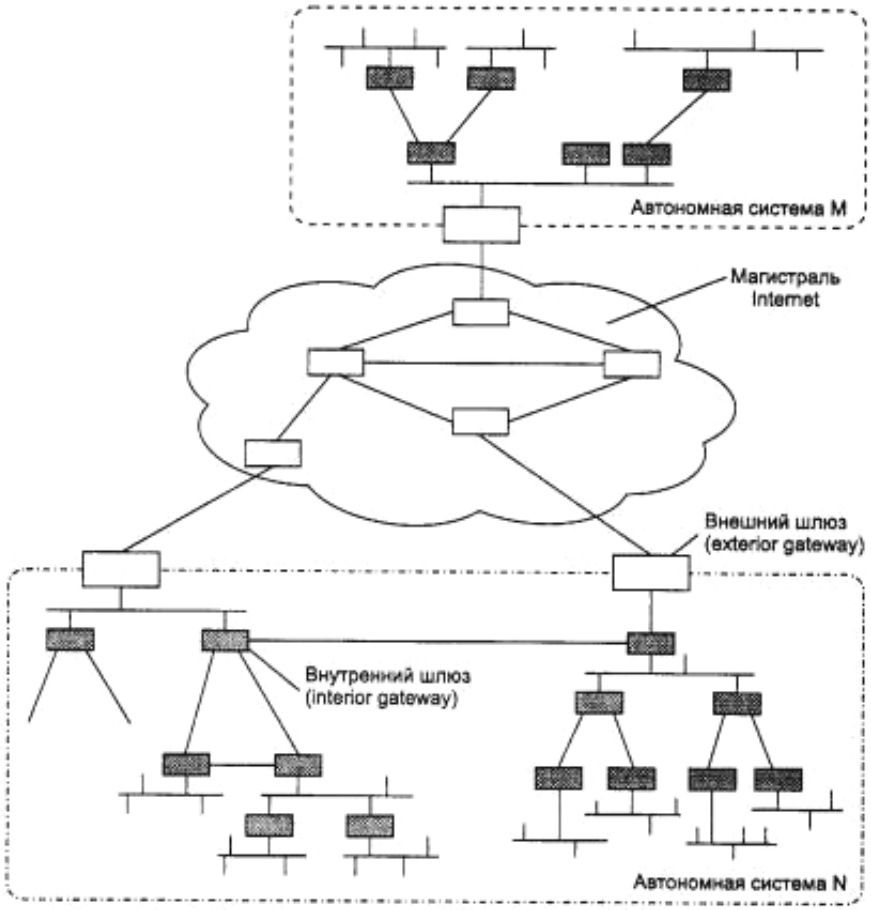


Рис. 2.97. Магистраль и автономные системы Internet

Соответственно протоколы маршрутизации внутри автономных систем называются *протоколами внутренних шлюзов (interior gateway protocol, IGP)*, а протоколы, определяющие обмен маршрутной информаци-

ей между внешними шлюзами и шлюзами магистральной сети, – *протоколами внешних шлюзов (exterior gateway protocol, EGP)*. Внутри магистральной сети также допустим любой собственный внутренний протокол IGP.

Смысл разделения всей сети Internet на автономные системы – в ее многоуровневом модульном представлении, что необходимо для любой крупной системы, способной к расширению в больших масштабах. Изменение протоколов маршрутизации внутри какой-либо автономной системы никак не должно влиять на работу остальных автономных систем. Кроме того, деление Internet на автономные системы должно способствовать агрегированию информации в магистральных и внешних шлюзах. Внутренние шлюзы могут использовать для внутренней маршрутизации достаточно подробные графы связей между собой, чтобы выбрать наиболее рациональный маршрут. Однако если информация такой степени детализации будет храниться во всех маршрутизаторах сети, то топологические базы данных так разрастутся, что потребуют наличия памяти гигантских размеров, а время принятия решений о маршрутизации станет неприемлемо большим.

Поэтому детальная топологическая информация остается внутри автономной системы, а автономную систему как единое целое для остальной части Internet представляют внешние шлюзы, которые сообщают о внутреннем составе автономной системы минимально необходимые сведения – количество IP-сетей, их адреса и внутреннее расстояние до этих сетей от данного внешнего шлюза.

Техника бесклассовой маршрутизации CIDR может значительно сократить объемы маршрутной информации, передаваемой между автономными системами. Так, если все сети внутри некоторой автономной системы начинаются с общего префикса, например 194.27.0.0/16, то внешний шлюз этой автономной системы должен делать объявления только об этом адресе, не сообщая отдельно о существовании внутри данной автономной системы, например, сети 194.27.32.0/19 или 194.27.40.0/21, т. к. эти адреса агрегируются в адрес 194.27.0.0/16.

Приведенная на рис. 2.97 структура Internet с единственной магистралью достаточно долго соответствовала действительности, поэтому специально для нее был разработан протокол обмена маршрутной информацией между автономными системами, названный EGP. Однако по мере развития сетей поставщиков услуг структура Internet стала гораздо более сложной, с произвольным характером связей между автономными системами. Поэтому протокол EGP уступил место протоколу BGP, который по-

зволяет распознать наличие петель между автономными системами и исключить их из межсистемных маршрутов. Протоколы EGP и BGP используются только во внешних шлюзах автономных систем, которые чаще всего организуются поставщиками услуг Internet. В маршрутизаторах корпоративных сетей работают внутренние протоколы маршрутизации, такие как RIP и OSPF.

Дистанционно-векторный протокол RIP

Построение таблицы маршрутизации

Протокол RIP (Routing Information Protocol) является внутренним протоколом маршрутизации дистанционно-векторного типа, он представляет собой один из наиболее ранних протоколов обмена маршрутной информацией и до сих пор чрезвычайно распространен в вычислительных сетях ввиду простоты реализации. Кроме версии RIP для сетей TCP/IP существует также версия RIP для сетей IPX/SPX компании Novell.

Для IP имеются две версии протокола RIP: первая и вторая. Протокол RIPvl не поддерживает масок, т. е. он распространяет между маршрутизаторами только информацию о номерах сетей и расстояниях до них, а информацию о масках этих сетей не распространяет, считая, что все адреса принадлежат к стандартными классам A, B или C. Протокол RIPv2 передает информацию о масках сетей, поэтому он в большей степени соответствует требованиям сегодняшнего дня. Так как при построении таблиц маршрутизации работа версии 2 принципиально не отличается от версии 1, то в дальнейшем для упрощения записей будет описываться работа первой версии.

В качестве расстояния до сети стандарты протокола RIP допускают различные виды метрик – хопы, метрики, учитывающие пропускную способность, вносимые задержки и надежность сетей (т. е. соответствующие признакам D, T и R в поле «Качество сервиса» IP-пакета), а также любые комбинации этих метрик. Метрика должна обладать свойством аддитивности – метрика составного пути должна быть равна сумме метрик составляющих этого пути. В большинстве реализаций RIP используется простейшая метрика – количество хопов, т. е. количество промежуточных маршрутизаторов, которые нужно преодолеть пакету до сети назначения.

Рассмотрим процесс построения таблицы маршрутизации с помощью протокола RIP на примере составной сети, изображенной на рис. 2.98.

Рис. 2.98. Сеть, объединенная RIP-маршрутизаторами

Этап 1. Создание минимальных таблиц. В этой сети имеются восемь IP-сетей, связанных четырьмя маршрутизаторами с идентификаторами M1, M2, M3 и M4. Маршрутизаторы, работающие по протоколу RIP, могут иметь идентификаторы, однако для работы протокола они не являются необходимыми. В RIP-сообщениях эти идентификаторы не передаются.

В исходном состоянии в каждом маршрутизаторе программным обеспечением стека TCP/IP автоматически создается минимальная таблица маршрутизации, в которой учитываются только непосредственно подсоединенные сети. На рисунке адреса портов маршрутизаторов, в отличие от адресов сетей, помещены в овалы.

Таблица 2.9 отражает примерный вид минимальной таблицы маршрутизации маршрутизатора M1.

Таблица 2.9

Минимальная таблица маршрутизации маршрутизатора Ml

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	201.36.14.3	1	1
132.11.0.0	132.11.0.7	2	1
194.27.18.0	194.27.18.1	3	1

Минимальные таблицы маршрутизации в других маршрутизаторах будут выглядеть соответственно, например, таблица маршрутизатора M2 будет состоять из трех записей (табл. 2.10).

Таблица 2.10

Минимальная таблица маршрутизации маршрутизатора М2

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
132.11.0.0	132.11.0.101	1	1
132.17.0.0	132.17.0.1	2	1
132.15.0.0	132.15.0.6	3	1

Этап 2. Рассылка минимальных таблиц соседям. После инициализации каждого маршрутизатора он начинает посылать своим соседям сообщения протокола RIP, в которых содержится его минимальная таблица.

RIP-сообщения передаются в пакетах протокола UDP и включают два параметра для каждой сети: ее IP-адрес и расстояние до нее от передающего сообщение маршрутизатора.

Соседями являются те маршрутизаторы, которым данный маршрутизатор непосредственно может передать IP-пакет по какой-либо своей сети, не пользуясь услугами промежуточных маршрутизаторов. Например, для маршрутизатора М1 соседями являются маршрутизаторы М2 и М3, а для маршрутизатора М4 – маршрутизаторы М2 и М3.

Таким образом, маршрутизатор М1 передает маршрутизатору М2 и М3 следующее сообщение:

сеть 201.36.14.0, расстояние 1;

сеть 132.11.0.0, расстояние 1;

сеть 194.27.18.0, расстояние 1.

Этап 3. Получение RIP-сообщений от соседей и обработка полученной информации. После получения аналогичных сообщений от маршрутизаторов М2 и М3 маршрутизатор М1 наращивает каждое полученное поле метрики на единицу и запоминает, через какой порт и от какого маршрутизатора получена новая информация (адрес этого маршрутизатора будет адресом следующего маршрутизатора, если эта запись будет внесена в таблицу маршрутизации). Затем маршрутизатор начинает сравнивать новую информацию с той, которая хранится в его таблице маршрутизации (табл. 2.11).

Записи с четвертой по девятую получены от соседних маршрутизаторов, и они претендуют на помещение в таблицу. Однако только записи с четвертой по седьмую попадают в таблицу, а записи восьмая и девятая – нет. Это происходит потому, что они содержат данные об уже имеющихся в таблице маршрутизатора М1 сетях, а расстояние до них хуже, чем в существующих записях.

Таблица 2.11

Таблица маршрутизации маршрутизатора М1

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	201.36.14.3	1	1
132.11.0.0	132.11.0.7	2	1
194.27.18.0	194.27.18.1	3	1
132.17.0.0	132.11.0.101	2	2
132.15.0.0	132.11.0.101	2	2
194.27.19.0	194.27.18.51	3	2
202.101.15.0	194.27.18.51	3	2
~~132.11.0.0~~	~~132.11.0.101~~	~~2~~	~~2~~
~~194.27.10.0~~	~~194.27.10.51~~	~~3~~	~~2~~

Протокол RIP замещает запись о какой-либо сети только в том случае, если новая информация имеет лучшую метрику (расстояние в хопах меньше), чем имеющаяся. В результате в таблице маршрутизации о каждой сети остается только одна запись; если же имеется несколько равнозначных в отношении расстояния путей к одной и той же сети, то все равно в таблице остается одна запись, которая пришла в маршрутизатор первая по времени. Для этого правила существует исключение – если худшая информация о какой-либо сети пришла от того же маршрутизатора, на основании сообщения которого была создана данная запись, то худшая информация замещает лучшую.

Аналогичные операции с новой информацией выполняют и остальные маршрутизаторы сети.

Этап 4. Рассылка новой, уже не минимальной, таблицы соседям. Каждый маршрутизатор отсылает новое RIP-сообщение всем своим соседям. В этом сообщении он помещает данные обо всех известных ему сетях, как непосредственно подключенных, так и удаленных, о которых маршрутизатор узнал из RIP-сообщений.

Этап 5. Получение RIP-сообщений от соседей и обработка полученной информации. Этап 5 повторяет этап 3 – маршрутизаторы принимают RIP-сообщения, обрабатывают содержащуюся в них информацию и на ее основании корректируют свои таблицы маршрутизации.

Рассмотрим это на примере маршрутизатора М1 (табл. 2.12).

Таблица 2.12

Таблица маршрутизации маршрутизатора М1

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	201.36.14.3	1	1
132.11.0.0	132.11.0.7	2	1
194.27.18.0	194.27.18.1	3	1
132.17.0.0	132.11.0.101	2	2
132.15.0.0	132.11.0.101	2	2
~~132.15.0.0~~	~~194.27.10.51~~	~~3~~	~~3~~
194.27.19.0	194.27.18.51	3	2
~~194.27.19.0~~	~~132.11.0.101~~	~~2~~	~~3~~
~~202.101.15.0~~	~~194.27.10.51~~	~~3~~	~~2~~
202.101.16.0	132.11.0.101	2	3
~~202.101.16.0~~	~~194.27.10.51~~	~~3~~	~~3~~

На этом этапе маршрутизатор М1 получил от маршрутизатора М3 информацию о сети 132.15.0.0, которую тот, в свою очередь, на предыдущем цикле работы получил от маршрутизатора М4. Маршрутизатор уже знает о сети 132.15.0.0, причем старая информация имеет лучшую метрику, чем новая, поэтому новая информация об этой сети отбрасывается.

О сети 202.101.16.0 маршрутизатор М1 узнает на этом этапе впервые, причем данные о ней приходят от двух соседей – от М2 и М3. Поскольку метрики в этих сообщениях указаны одинаковые, то в таблицу попадают данные, которые пришли первыми. В нашем примере считается, что маршрутизатор М2 опередил маршрутизатор М3 и первым переслал свое RIP-сообщение маршрутизатору М1.

Если маршрутизаторы периодически повторяют этапы рассылки и обработки RIP-сообщений, то за конечное время в сети установится корректный режим маршругизации. Под корректным режимом маршрутизации здесь понимается такое состояние таблиц маршрутизации, когда все сети будут достижимы из любой сети с помощью некоторого рационального маршрута. Пакеты будут доходить до адресатов и не зацикливаться в петлях, подобных той, которая образуется на рис. 2.98 маршрутизаторами М1-М2-М3-М4.

Очевидно, если в сети все маршрутизаторы, их интерфейсы и соединяющие их каналы связи постоянно работоспособны, то объявления по протоколу RIP можно делать достаточно редко, например, один раз в день. Однако в сетях постоянно происходят изменения – изменяется работоспособность маршрутизаторов и каналов, и сами маршрутизаторы и каналы могут добавляться в существующую сеть или же выводиться из ее состава.

Для адаптации к изменениям в сети протокол RIP использует ряд механизмов.

Адаптация RIP-маршрутизаторов к изменениям состояния сети

К новым маршрутам RIP-маршрутизаторы приспосабливаются просто – они передают новую информацию в очередном сообщении своим соседям и постепенно эта информация становится известна всем маршрутизаторам сети. А вот к отрицательным изменениям, связанным с потерей какого-либо маршрута, RIP-маршрутизаторы приспосабливаются сложнее. Это связано с тем, что в формате сообщений протокола RIP нет поля, которое бы указывало на то, что путь к данной сети больше не существует.

Вместо этого используются два механизма уведомления о том, что некоторый маршрут более недействителен:

– истечение времени жизни маршрута;

– указание специального расстояния (бесконечности) до сети, ставшей недоступной.

Для отработки первого механизма каждая запись таблицы маршрутизации, полученная по протоколу RIP, имеет время жизни (TTL). При поступлении очередного RIP-сообщения, которое подтверждает справедливость данной записи, таймер TTL устанавливается в исходное состояние, а затем из него каждую секунду вычитается единица. Если за время тайм-аута не придет новое маршрутное сообщение об этом маршруте, то он помечается как недействительный. В RIP период рассылки выбран равным 30 секундам, а в качестве тайм-аута выбрано шестикратное значение периода рассылки, т. е. 180 секунд.

Тайм-аут работает в тех случаях, когда маршрутизатор не может послать соседям сообщение об отказавшем маршруте, т. к. либо он сам неработоспособен, либо неработоспособна линия связи, по которой можно было бы передать сообщение.

Когда же сообщение послать можно, RIP-маршрутизаторы не используют специальный признак в сообщении, а указывают бесконечное расстояние до сети, причем в протоколе RIP оно выбрано равным 16 хопам (при другой метрике необходимо указать маршрутизатору ее значение, считающееся бесконечностью). Получив сообщение, в котором некоторая сеть сопровождается расстоянием 16 (или 15, что приводит к тому же результату, т. к. маршрутизатор наращивает полученное значение на 1), маршрутизатор должен проверить, исходит ли эта «плохая» информация о сети от того же маршрутизатора, сообщение которого послужило в свое

время основанием для записи о данной сети в таблице маршрутизации. Если это тот же маршрутизатор, то информация считается достоверной и маршрут помечается как недоступный.

Такое небольшое значение «бесконечного» расстояния вызвано тем, что в некоторых случаях отказы связей в сети вызывают длительные периоды некорректной работы RIP-маршрутизаторов, выражающейся в зацикливании пакетов в петлях сети. И чем меньше расстояние, используемое в качестве «бесконечного», тем такие периоды становятся короче.

Рассмотрим случай зацикливания пакетов на примере сети, изображенной на рис. 2.98.

Пусть маршрутизатор М1 обнаружил, что его связь с непосредственно подключенной сетью 201.36.14.0 потеряна (например, по причине отказа интерфейса 201.36.14.3). М1 отметил в своей таблице маршрутизации, что сеть 201.36.14.0 недоступна. В худшем случае он обнаружил это сразу же после отправки очередных RIP-сообщений, так что до начала нового цикла его объявлений, в котором он должен сообщить соседям, что расстояние до сети 201.36.14.0 стало равным 16, остается почти 30 секунд.

Каждый маршрутизатор работает на основании своего внутреннего таймера, не синхронизируя работу по рассылке объявлений с другими маршрутизаторами. Поэтому весьма вероятно, что маршрутизатор М2 опередил маршрутизатор М1 и передал ему свое сообщение раньше, чем М1 успел передать новость о недостижимости сети 201.36.14.0. А в этом сообщении имеются данные, порожденные следующей записью в таблице маршрутизации М2 (табл. 2.13).

Таблица 2.13

Таблица маршрутизации маршрутизатора М2

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	132.11.0.7	1	2

Эта запись была получена от маршрутизатора М1 и корректна до отказа интерфейса 201.36.14.3, а теперь она устарела, но маршрутизатор М2 об этом не узнал.

Теперь маршрутизатор М1 получил новую информацию о сети 201.36.14.0 – эта сеть достижима через маршрутизатор М2 с метрикой 2. Раньше М1 также получал эту информацию от М2. Но игнорировал ее, так как его собственная метрика для 201.36.14.0 была лучше. Теперь М1 должен принять данные о сети 201.36.14.0, полученные от М2, и заменить запись в таблице маршрутизации о недостижимости этой сети (табл. 2.14).

Таблица 2.14

Таблица маршрутизации маршрутизатора М1

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	132.11.0.101	2	3

В результате в сети образовалась маршрутная петля – пакеты, направляемые узлам сети 201.36.14.0, будут передаваться маршрутизатором М2 маршрутизатору М1, а маршрутизатор М1 будет возвращать их маршрутизатору М2. IP-пакеты будут циркулировать по этой петле до тех пор, пока не истечет время жизни каждого пакета.

Маршрутная петля будет существовать в сети достаточно долго. Рассмотрим периоды времени, кратные времени жизни записей в таблицах маршрутизаторов.

Время 0 – 180 с. После отказа интерфейса в маршрутизаторах М1 и М2 будут сохраняться некорректные записи, приведенные выше. Маршрутизатор М2 по-прежнему снабжает маршрутизатор М1 своей записью о сети 201.36.14.0 с метрикой 2, т. к. ее время жизни не истекло. Пакеты зацикливаются.

Время 180 – 360 с. В начале этого периода у маршрутизатора М2 истекает время жизни записи о сети 201.36.14.0 с метрикой 2, т. к. маршрутизатор М1 в предыдущий период посылал ему сообщения о сети 201.36.14.0 с худшей метрикой, чем у М2, и они не могли подтверждать эту запись. Теперь маршрутизатор М2 принимает от маршрутизатора М1 запись о сети 201.36.14.0 с метрикой 3 и трансформирует ее в запись с метрикой 4. Маршрутизатор М1 не получает новых сообщений от маршрутизатора М2 о сети 201.36.14.0 с метрикой 2, поэтому время жизни его записи начинает уменьшаться. Пакеты продолжают зацикливаться.

Время 360 – 540 с. Теперь у маршрутизатора М1 истекает время жизни записи о сети 201.36.14.0 с метрикой 3. Маршрутизаторы М1 и М2 опять меняются ролями – М2 снабжает М1 устаревшей информацией о пути к сети 201.36.14.0, уже с метрикой 4, которую М1 преобразует в метрику 5. Пакеты продолжают зацикливаться.

Если бы в протоколе RIP не было выбрано расстояние 16 в качестве недостижимого, то описанный процесс длился бы до бесконечности (вернее, пока не была бы исчерпана разрядная сетка поля расстояния и не было бы зафиксировано переполнения при очередном наращивании расстояния).

В результате маршрутизатор М2 на очередном этапе описанного процесса получает от маршрутизатора М1 метрику 15, которая после наращи-

вания, превращаясь в метрику 16, фиксирует недостижимость сети. Период нестабильной работы сети длился 36 минут!

Ограничение в 15 хопов сужает область применения протокола RIP до сетей, в которых число промежуточных маршрутизаторов не может быть больше 15. Для более масштабных сетей нужно применять другие протоколы маршрутизации, например OSPF, или разбивать сеть на автономные области.

Приведенный пример хорошо иллюстрирует главную причину нестабильной работы маршрутизаторов, работающих по протоколу RIP. Эта причина коренится в самом принципе работы дистанционно-векторных протоколов – пользовании информацией, полученной из вторых рук. Действительно, маршрутизатор М2 передал маршрутизатору М1 информацию о достижимости сети 201.36.14.0, за достоверность которой он сам не отвечает. Устранить эту причину полностью нельзя, ведь сам способ построения таблиц маршрутизации связан с передачей чужой информации без указания источника ее происхождения.

Методы борьбы с ложными маршрутами в протоколе RIP

Несмотря на то, что протокол RIP не в состоянии полностью исключить переходные состояния в сети, когда некоторые маршрутизаторы пользуются устаревшей информацией об уже несуществующих маршрутах, имеется несколько методов, которые во многих случаях решают подобные проблемы.

Ситуация с петлей, образующейся между соседними маршрутизаторами, описанная выше, надежно решается с помощью метода, получившего название *расщепление горизонта (split horizon)*. Метод заключается в том, что маршрутная информация о некоторой сети, хранящаяся в таблице маршрутизации, никогда не передается тому маршрутизатору, от которого она получена (это следующий маршрутизатор в данном маршруте). Если маршрутизатор М2 в рассмотренном выше примере поддерживает технику расщепления горизонта, то он не передаст маршрутизатору М1 устаревшую информацию о сети 201.36.14.0, т. к. получил ее именно от маршрутизатора М1.

Практически все сегодняшние маршрутизаторы, работающие по протоколу RIP, используют технику расщепления горизонта.

Однако расщепление горизонта не помогает в тех случаях, когда петли образуются не двумя, а несколькими маршрутизаторами. Рассмотрим более детально ситуацию, которая возникнет в сети, приведенной на рис. 2.98, в случае потери связи маршрутизатора М1 с сетью 201.36.14.0.

Пусть все маршрутизаторы этой сети поддерживают технику расщепления горизонта. Маршрутизаторы М2 и М3 не будут возвращать маршрутизатору в этой ситуации данные о сети 201.36.14.0 с метрикой 2, т. к. они получили эту информацию от маршрутизатора М1. Однако они будут передавать маршрутизатору информацию о достижимости сети 201.36.14.0 с метрикой 4 через себя, т. к. получили эту информацию по сложному маршруту, а не от маршрутизатора М1 непосредственно. Например, маршрутизатор М2 получил эту информацию по цепочке М4-М3-М1. Поэтому маршрутизатор М1 снова может быть обманут, пока каждый из маршрутизаторов в цепочке М3-М4-М2 не вычеркнет запись о достижимости сети 1 (а это произойдет через период 3 × 180 секунд).

Для предотвращения зацикливания пакетов по составным петлям при отказах связей применяются два других приема, называемые *триггерными обновлениями (triggered updates)* и *замораживанием изменений (hold down).*

Способ триггерных обновлений состоит в том, что маршрутизатор, получив данные об изменении метрики до какой-либо сети, не ждет истечения периода передачи таблицы маршрутизации, а передает данные об изменившемся маршруте немедленно. Этот прием может во многих случаях предотвратить передачу устаревших сведений об отказавшем маршруте, но он перегружает сеть служебными сообщениями, поэтому триггерные объявления также делаются с некоторой задержкой. Возможна ситуация, когда регулярное обновление в каком-либо маршрутизаторе чуть опередит по времени приход триггерного обновления от предыдущего в цепочке маршрутизатора и данный маршрутизатор успеет передать по сети устаревшую информацию о несуществующем маршруте.

Второй прием позволяет исключить подобные ситуации. Он связан с введением тайм-аута на принятие новых данных о сети, которая только что стала недоступной. Этот тайм-аут предотвращает принятие устаревших сведений о некотором маршруте от тех маршрутизаторов, которые находятся на некотором расстоянии от отказавшей связи и передают устаревшие сведения о ее работоспособности. Предполагается, что в течение тайм-аута «замораживания изменений» эти маршрутизаторы вычеркнут данный маршрут из своих таблиц, т. к. не получат о нем новых записей и не будут распространять устаревшие сведения по сети.

Протокол «состояния связей» OSPF

Протокол *OSPF (Open Shortest Path First)* является достаточно современной реализацией алгоритма состояния связей (он принят в 1991 го-

ду) и обладает многими особенностями, ориентированными на применение в больших гетерогенных сетях.

В OSPF процесс построения таблицы маршрутизации разбивается на два крупных этапа:

– на первом этапе каждый маршрутизатор строит граф связей сети, в котором вершинами графа являются маршрутизаторы и IP-сети, а ребрами – интерфейсы маршрутизаторов. Все маршрутизаторы для этого обмениваются со своими соседями той информацией о графе сети, которой они располагают к данному моменту времени. Этот процесс похож на процесс распространения векторов расстояний до сетей в протоколе RIP, однако сама информация качественно другая – это информация о топологии сети. Эти сообщения называются *router links advertisement* – объявление о связях маршрутизатора. Кроме того, при передаче топологической информации маршрутизаторы ее не модифицируют, как это делают RIP-маршрутизаторы, а передают в неизменном виде. В результате распространения топологической информации все маршрутизаторы сети располагают идентичными сведениями о графе сети, которые хранятся в топологической базе данных маршрутизатора;

– второй этап состоит в нахождении оптимальных маршрутов с помощью полученного графа. Каждый маршрутизатор считает себя центром сети и ищет оптимальный маршрут до каждой известной ему сети. В каждом найденном таким образом маршруте запоминается только один шаг – до следующего маршрутизатора, в соответствии с принципом одношаговой маршрутизации. Данные об этом шаге и попадают в таблицу маршрутизации. Задача нахождения оптимального пути на графе является достаточно сложной и трудоемкой. В протоколе OSPF для ее решения используется итеративный алгоритм Дейкстры. Если несколько маршрутов имеют одинаковую метрику до сети назначения, то в таблице маршрутизации запоминаются первые шаги всех этих маршрутов.

После первоначального построения таблицы маршрутизации необходимо отслеживать изменения состояния сети и вносить коррективы в таблицу маршрутизации. Для контроля состояния связей и соседних маршрутизаторов OSPF-маршрутизаторы не используют обмен полной таблицей маршрутизации. Вместо этого они передают специальные короткие сообщения HELLO. Если состояние сети не меняется, то OSPF-маршрутизаторы корректировкой своих таблиц маршрутизации не занимаются и не посылают соседям объявления о связях. Если же состояние связи изменилось, то ближайшим соседям посылается новое объявление, касающееся только данной связи, что, конечно, экономит пропускную спо-

собность сети. Получив новое объявление об изменении состояния связи, маршрутизатор перестраивает граф сети, заново ищет оптимальные маршруты (не обязательно все, а только те, на которых отразилось данное изменение) и корректирует свою таблицу маршрутизации. Одновременно маршрутизатор ретранслирует объявление каждому из своих ближайших соседей (кроме того, от которого он получил это объявление).

При появлении новой связи или нового соседа маршрутизатор узнает об этом из новых сообщений HELLO. В сообщениях HELLO указывается достаточно детальная информация о том маршрутизаторе, который послал это сообщение, а также о его ближайших соседях, чтобы данный маршрутизатор можно было однозначно идентифицировать. Сообщения HELLO отправляются через каждые 10 секунд, чтобы повысить скорость адаптации маршрутизаторов к изменениям, происходящим в сети. Небольшой объем этих сообщений делает возможным такое частое тестирование состояния соседей и связей с ними.

Так как маршрутизаторы являются одними из вершин графа, то они обязательно должны иметь идентификаторы.

Протокол OSPF обычно использует метрику, учитывающую пропускную способность сетей. Кроме того, возможно использование двух других метрик, учитывающих требования к качеству обслуживания в IP-пакете, – задержки передачи пакетов и надежности передачи пакетов сетью. Для каждой из метрик протокол OSPF строит отдельную таблицу маршрутизации. Выбор нужной таблицы происходит в зависимости от требований к качеству обслуживания пришедшего пакета (рис. 2.99).

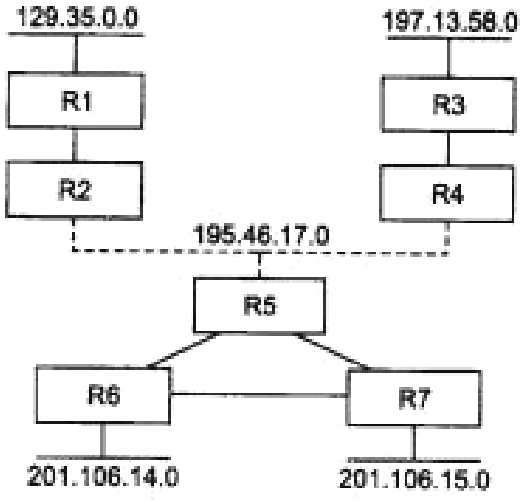


Рис. 2.99. Построение таблицы маршрутизации по протоколу OSPF

Маршрутизаторы соединены как с локальными сетями, так и непосредственно между собой глобальными каналами типа «точка – точка».

Данной сети соответствует граф, приведенный на рис. 2.100.

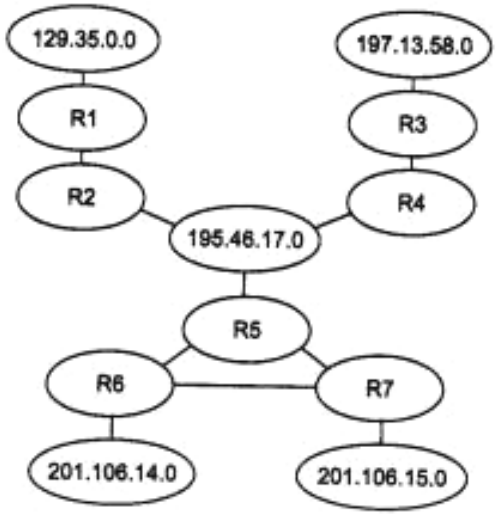


Рис. 2.100. Граф сети, построенный протоколом OSPF

Протокол OSPF в своих объявлениях распространяет информацию о связях двух типов: «маршрутизатор – маршрутизатор» и «маршрутизатор – сеть». Примером связи первого типа служит связь «R3 – R4», а второго – связь «R4 – 195.46.17.0». Если каналам «точка – точка» дать IP-адреса, то они станут дополнительными вершинами графа, как и локальные сети. Вместе с IP-адресом сети передается также информация о маске сети.

После инициализации OSPF-маршрутизаторы знают только о связях с непосредственно подключенными сетями, как и RIP-маршрутизаторы. Они начинают распространять эту информацию своим соседям. Одновременно они посылают сообщения HELLO по всем своим интерфейсам, так что почти сразу же маршрутизатор узнает идентификаторы своих ближайших соседей, что пополняет его топологическую базу новой информацией, которую он узнал непосредственно. Далее топологическая информация начинает распространяться по сети от соседа к соседу и через некоторое время достигает самых удаленных маршрутизаторов.

Каждая связь характеризуется метрикой. Протокол OSPF поддерживает стандартные для многих протоколов (например, для протокола Spanning Tree) значения расстояний для метрики, отражающей производительность сетей: Ethernet – 10 единиц, Fast Ethernet – 1 единица, канал T1 – 65 единиц, канал 56 Кбит/с – 1785 единиц и т. д.

При выборе оптимального пути на графе с каждым ребром графа связана метрика, которая добавляется к пути, если данное ребро в него

входит. Пусть на приведенном примере маршрутизатор R5 связан с R6 и R7 каналами T1, а R6 и R7 связаны между собой каналом 56 Кбит/с. Тогда R7 определит оптимальный маршрут до сети 201.106.14.0 как составной, проходящий сначала через маршрутизатор R5, а затем через R6, поскольку у этого маршрута метрика будет равна 65 + 65 = 130 единиц. Непосредственный маршрут через R6 не будет оптимальным, т. к. его метрика равна 1785. При использовании хопов был бы выбран маршрут через R6, что не было бы оптимальным.

Протокол OSPF разрешает хранить в таблице маршрутизации несколько маршрутов к одной сети, если они обладают равными метриками. Если такие записи образуются в таблице маршрутизации, то маршрутизатор реализует режим баланса загрузки маршрутов (*load balancing*), отправляя пакеты попеременно по каждому из маршрутов.

У каждой записи в топологической базе данных имеется срок жизни, как и у маршрутных записей протокола RIP. С каждой записью о связях связан таймер, который используется для контроля времени жизни записи. Если какая-либо запись топологической базы маршрутизатора, полученная от другого маршрутизатора, устаревает, то он может запросить ее новую копию с помощью специального сообщения Link-State Request протокола OSPF, на которое должен поступить ответ Link-State Update от маршрутизатора, непосредственно тестирующего запрошенную связь.

При инициализации маршрутизаторов, а также для более надежной синхронизации топологических баз маршрутизаторы периодически обмениваются всеми записями базы, но этот период существенно больше, чем у RIP-маршрутизаторов.

Так как информация о некоторой связи изначально генерируется только тем маршрутизатором, который выяснил фактическое состояние этой связи путем тестирования с помощью сообщений HELLO, а остальные маршрутизаторы только ретранслируют эту информацию без преобразования, то недостоверная информация о достижимости сетей, которая может появляться в RIP-маршрутиэаторах, в OSPF-маршрутизаторах появиться не может, а устаревшая информация быстро заменяется новой, т. к. при изменении состояния связи новое сообщение генерируется сразу же.

В OSPF-сетях могут возникать периоды нестабильной работы. Например, при отказе связи, когда информация об этом не дошла до какого-либо маршрутизатора и он отправляет пакеты сети назначения, считая эту связь работоспособной. Однако эти периоды продолжаются недолго, причем пакеты не зацикливаются в маршрутных петлях, а просто отбрасываются при невозможности их передать через неработоспособную связь.

К недостаткам протокола OSPF следует отнести его вычислительную сложность, которая быстро растет с увеличением размерности сети, т. е. количества сетей, маршрутизаторов и связей между ними. Для преодоления этого недостатка в протоколе OSPF вводится понятие *область сети (area)*. Маршрутизаторы, принадлежащие некоторой области, строят граф связей только для этой области, что сокращает размерность сети. Между областями информация о связях не передается, а пограничные для областей маршрутизаторы обмениваются только информацией об адресах сетей, имеющихся в каждой из областей, и расстоянием от пограничного маршрутизатора до каждой сети. При передаче пакетов между областями выбирается один из пограничных маршрутизаторов области, а именно тот, у которого расстояние до нужной сети меньше. Этот стиль напоминает стиль работы протокола RIP, но нестабильность здесь устраняется тем, что петлевидные связи между областями запрещены. При передаче адресов в другую область OSPF-маршрутизаторы агрегируют несколько адресов в один, если обнаруживают у них общий префикс.

OSPF-маршрутизаторы могут принимать адресную информацию от других протоколов маршрутизации, например от протокола RIP, что полезно для работы в гетерогенных сетях. Такая адресная информация обрабатывается так же, как и внешняя информация между разными областями.

2.30. Основные характеристики маршрутизаторов. Стирание граней между маршрутизаторами и коммутаторами

Основная задача маршрутизатора – выбор наилучшего маршрута в сети – часто является достаточно сложной с математической точки зрения. Особенно интенсивных вычислений требуют протоколы, основанные на алгоритме состояния связей, вычисляющие оптимальный путь на графе, – OSPF, NLSP, IS-IS. Кроме этой основной функции в круг ответственности маршрутизатора входят и другие задачи, такие как буферизация, фильтрация и фрагментация перемещаемых пакетов. При этом очень важна производительность, с которой маршрутизатор выполняет эти задачи.

Поэтому типичный маршрутизатор является мощным вычислительным устройством с одним или даже несколькими процессорами, часто специализированными или построенными на RISC-архитектуре, со сложным программным обеспечением. То есть сегодняшний маршрутизатор – это специализированный компьютер, имеющий скоростную внутреннюю шину или несколько шин, часто использующий симметричное или асимметричное мультипроцессирование и работающий под управлением спе-

циализированной операционной системы, относящейся к классу систем реального времени.

Маршрутизаторы могут поддерживать как один протокол сетевого уровня (например, IP, IPX или DECnet), так и множество таких протоколов. В последнем случае они называются многопротокольными маршрутизаторами. Чем больше протоколов сетевого уровня поддерживает маршрутизатор, тем лучше он подходит для корпоративной сети.

Большая вычислительная мощность позволяет маршрутизаторам наряду с основной работой по выбору оптимального маршрута выполнять и ряд вспомогательных высокоуровневых функций.

Классификация маршрутизаторов по областям применения

По областям применения маршрутизаторы делятся на несколько классов.

Магистральные маршрутизаторы (backbone routers) предназначены для построения центральной сети корпорации. Центральная сеть может состоять из большого количества локальных сетей, разбросанных по разным зданиям и использующих самые разнообразные сетевые технологии, типы компьютеров и операционных систем. Магистральные маршрутизаторы – это наиболее мощные устройства, способные обрабатывать несколько сотен тысяч или даже несколько миллионов пакетов в секунду, имеющие большое количество интерфейсов локальных и глобальных сетей. Чаще всего магистральный маршрутизатор конструктивно выполнен по модульной схеме на основе шасси с большим количеством слотов – до 12 – 14. Большое внимание уделяется в магистральных моделях надежности и отказоустойчивости маршрутизатора, которая достигается за счет системы терморегуляции, избыточных источников питания, заменяемых «на ходу» (*hot swap*) модулей, а также симметричного мультипроцессирования.

Маршрутизаторы региональных отделений соединяют региональные отделения между собой и с центральной сетью. Сеть регионального отделения, так же как и центральная сеть, может состоять из нескольких локальных сетей. Такой маршрутизатор обычно представляет собой некоторую упрощенную версию магистрального маршрутизатора. Если он выполнен на основе шасси, то количество слотов его шасси меньше: 4 – 5. Возможен также конструктив с фиксированным количеством портов. Поддерживаемые интерфейсы локальных и глобальных сетей менее скоростные.

Маршрутизаторы удаленных офисов соединяют, как правило, единственную локальную сеть удаленного офиса с центральной сетью или сетью регионального отделения по глобальной связи. Маршрутизатор удаленного офиса может поддерживать работу по коммутируемой телефонной линии в качестве резервной связи для выделенного канала. Существует очень большое количество типов маршрутизаторов удаленных офисов. Это объясняется как массовостью потенциальных потребителей, так и специализацией такого типа устройств, проявляющейся в поддержке одного конкретного типа глобальной связи. Например, существуют маршрутизаторы, работающие только по сети ISDN, существуют модели только для аналоговых выделенных линий и т. п.

Маршрутизаторы локальных сетей (коммутаторы 3-го уровня) предназначены для разделения крупных локальных сетей на подсети. Основное требование, предъявляемое к ним, – высокая скорость маршрутизации, т. к. в такой конфигурации отсутствуют низкоскоростные порты, такие как модемные порты 33,6 Кбит/с или цифровые порты 64 Кбит/с.

В зависимости от области применения маршрутизаторы обладают различными основными и дополнительными техническими характеристиками.

Основные технические характеристики маршрутизатора

Основные технические характеристики маршрутизатора связаны с тем, как он решает свою главную задачу, – маршрутизацию пакетов в составной сети. Именно эти характеристики прежде всего определяют возможности и сферу применения того или иного маршрутизатора.

Перечень поддерживаемых сетевых протоколов. Магистральный маршрутизатор должен поддерживать большое количество сетевых протоколов и протоколов маршрутизации, чтобы обеспечивать трафик всех существующих на предприятии вычислительных систем (в том числе и устаревших, но все еще успешно эксплуатирующихся, так называемых унаследованных – *legacy*), а также систем, которые могут появиться на предприятии в ближайшем будущем. Если центральная сеть образует отдельную автономную систему Internet, то потребуется поддержка и специфических протоколов маршрутизации этой сети, таких как EGP и BGP. Программное обеспечение магистральных маршрутизаторов обычно строится по модульному принципу, поэтому при возникновении потребности можно докупать и добавлять программные модули, реализующие недостающие протоколы.

Перечень поддерживаемых интерфейсов локальных и глобальных сетей. Для локальных сетей это интерфейсы, реализующие физические и канальные протоколы сетей Ethernet, Token Ring, FDDI, Fast Ethernet, Gigabit Ethernet, 100VG-AnyLAN иATM.

Для глобальных связей это интерфейсы физического уровня для связи с аппаратурой передачи данных, а также протоколы канального и сетевого уровней, необходимые для подключения к глобальным сетям с коммутацией каналов и пакетов.

Поддерживаются интерфейсы последовательных линий (*serial lines*) RS-232, RS-449/422, V.35 (для передачи данных со скоростями до 2 – 6 Мбит/с), высокоскоростной интерфейс HSSI, обеспечивающий скорость до 52 Мбит/с, а также интерфейсы с цифровыми каналами T1/E1, T3/E3 и интерфейсами BRI и PRI цифровой сети ISDN. Некоторые маршрутизаторы имеют аппаратуру связи с цифровыми глобальными каналами, что исключает необходимость использования внешних устройств сопряжения с этими каналами.

В набор поддерживаемых глобальных технологий обычно входят технологии X.25, frame relay, ISDN и коммутируемых аналоговых телефонных сетей, сетей ATM, а также поддержка протокола канального уровня PPP.

Общая производительность маршрутизатора. Высокая производительность маршрутизации важна для работы с высокоскоростными локальными сетями, а также для поддержки новых высокоскоростных глобальных технологий, таких как frame relay, T3/E3, SDH и ATM. Общая производительность маршрутизатора зависит от многих факторов, наиболее важными из которых являются тип используемых процессоров, эффективность программной реализации протоколов, архитектурная организация вычислительных и интерфейсных модулей. Общая производительность маршрутизаторов колеблется от нескольких десятков тысяч пакетов в секунду до нескольких миллионов пакетов в секунду. Наиболее производительные маршрутизаторы имеют мультипроцессорную архитектуру, сочетающую симметричные и асимметричные свойства, – несколько мощных центральных процессоров по симметричной схеме выполняют функции вычисления таблицы маршрутизации, а менее мощные процессоры в интерфейсных модулях занимаются передачей пакетов на подключенные к ним сети и пересылкой пакетов на основании части таблицы маршрутизации, кэшированной в локальной памяти интерфейсного модуля.

Магистральные маршрутизаторы обычно поддерживают максимальный набор протоколов и интерфейсов и обладают высокой общей произво-

дительностью в один – два миллиона пакетов в секунду. Маршрутизаторы удаленных офисов поддерживают один – два протокола локальных сетей и низкоскоростные глобальные протоколы, общая производительность таких маршрутизаторов обычно составляет от 5 до 20 – 30 тысяч пакетов в секунду.

Маршрутизаторы региональных отделений занимают промежуточное положение, поэтому их иногда не выделяют в отдельный класс устройств.

Наиболее высокой производительностью обладают коммутаторы 3-го уровня, особенности которых рассмотрены ниже.

Дополнительные функциональные возможности маршрутизаторов

Наряду с функцией маршрутизации многие маршрутизаторы обладают следующими важными дополнительными функциональными возможностями, которые значительно расширяют сферу применения этих устройств.

Поддержка одновременно нескольких протоколов маршрутизации. В протоколах маршрутизации обычно предполагается, что маршрутизатор строит свою таблицу на основе работы только этого одного протокола. Деление Internet на автономные системы также направлено на исключение использования в одной автономной системе нескольких протоколов маршрутизации. Тем не менее, иногда в большой корпоративной сети приходится поддерживать одновременно несколько таких протоколов, чаще всего это складывается исторически. При этом таблица маршрутизации может получаться противоречивой – разные протоколы маршрутизации могут выбрать разные следующие маршрутизаторы для какой-либо сети назначения. Большинство маршрутизаторов решают эту проблему за счет придания приоритетов решениям разных протоколов маршрутизации. Высший приоритет отдается статическим маршрутам, следующий приоритет имеют маршруты, выбранные протоколами состояния связей, такими как OSPF или NLSP, а низшим приоритетом обладают маршруты дистанционно-векторных протоколов, как самых несовершенных.

Приоритеты сетевых протоколов. Можно установить приоритет одного протокола сетевого уровня над другими. На выбор маршрутов эти приоритеты не оказывают никакого влияния, они влияют только на порядок, в котором многопротокольный маршрутизатор обслуживает пакеты

разных сетевых протоколов. Это свойство бывает полезно в случае недостаточной полосы пропускания кабельной системы и существования трафика, чувствительного к временным задержкам, например трафика SNA или голосового трафика, передаваемого одним из сетевых протоколов.

Поддержка политики маршрутных объявлений. В большинстве протоколов обмена маршрутной информацией (RIP, OSPF, NLSP) предполагается, что маршрутизатор объявляет в своих сообщениях обо всех сетях, которые ему известны. Аналогично предполагается, что маршрутизатор при построении своей таблицы учитывает все адреса сетей, которые поступают ему от других маршрутизаторов сети. Однако существуют ситуации, когда администратор хотел бы скрыть существование некоторых сетей в определенной части своей сети от других администраторов, например, по соображениям безопасности. Или же администратор хотел бы запретить некоторые маршруты, которые могли бы существовать в сети. При статическом построении таблиц маршрутизации решение таких проблем не составляет труда. Динамические же протоколы маршрутизации не позволяют стандартным способом реализовывать подобные ограничения. Существует только один широко используемый протокол динамической маршрутизации, в котором описана возможность существования правил (*policy*), ограничивающих распространение некоторых адресов в объявлениях, – это протокол BGP. Необходимость поддержки таких правил в протоколе BGP понятна, т. к. это протокол обмена маршрутной информацией между автономными системами, где велика потребность в административном регулировании маршрутов (например, некоторый поставщик услуг Internet может не захотеть, чтобы через него транзитом проходил трафик другого поставщика услуг). Разработчики маршрутизаторов исправляют этот недостаток стандартов протоколов, вводя в маршрутизаторы поддержку правил передачи и использования маршрутной информации, подобных тем, которые рекомендует BGP.

Защита от широковещательных штормов (broadcast storm). Одна из характерных неисправностей сетевого программного обеспечения – самопроизвольная генерация с высокой интенсивностью широковещательных пакетов. Широковещательным штормом считается ситуация, в которой процент широковещательных пакетов превышает 20 % от общего количества пакетов в сети. Обычный коммутатор или мост слепо передает такие пакеты на все свои порты, как того требует его логика работы, засоряя, таким образом, сеть. Борьба с широковещательным штормом в сети, соединенной коммутаторами, требует от администратора отключения портов,

генерирующих широковещательные пакеты. Маршрутизатор не распространяет такие поврежденные пакеты, поскольку в круг его задач не входит копирование широковещательных пакетов во все объединяемые им сети. Поэтому маршрутизатор является хорошим средством борьбы с широковещательным штормом.

Поддержка немаршрутизируемых протоколов, таких как NetBIOS, NetBEUI или DEC LAT, которые не оперируют с таким понятием, как сеть. Маршрутизаторы могут обрабатывать пакеты таких протоколов двумя способами:

– в первом случае они могут работать с пакетами этих протоколов как мосты, т. е. передавать их на основании изучения MAC-адресов. Маршрутизатор необходимо сконфигурировать особым способом, чтобы по отношению к некоторым немаршрутизируемым протоколам на некоторых портах он выполнял функции моста, а по отношению к маршрутизируемым протоколам – функции маршрутизатора. Такой мост/маршрутизатор иногда называют *brouter* (*bridge* плюс *router*);

– другим способом передачи пакетов немаршрутизируемых протоколов является инкапсуляция этих пакетов в пакеты какого-либо сетевого протокола. Некоторые производители маршрутизаторов разработали собственные протоколы, специально предназначенные для инкапсуляции немаршрутизируемых пакетов. Кроме того, существуют стандарты для инкапсуляции некоторых протоколов в другие, в основном, в IP. Примером такого стандарта является протокол DLSw, определяющий методы инкапсуляции пакетов SDLC и NetBIOS в IP-пакеты, а также протоколы PPTP и L2TP, инкапсулирующие кадры протокола PPP в IP-пакеты.

Разделение функций построения и использования таблицы маршрутизации. Основная вычислительная работа проводится маршрутизатором при составлении таблицы маршрутизации с маршрутами ко всем известным ему сетям. Эта работа состоит в обмене пакетами протоколов маршрутизации, такими как RIP или OSPF, и вычислении оптимального пути к каждой целевой сети по некоторому критерию. Для вычисления оптимального пути на графе, как того требуют протоколы состояния связей, необходимы значительные вычислительные мощности. После того как таблица маршрутизации составлена, функция продвижения пакетов происходит весьма просто – осуществляется просмотр таблицы и поиск совпадения полученного адреса с адресом целевой сети. Если совпадение есть, то пакет передается на соответствующий порт маршрутизатора. Некоторые маршрутизаторы поддерживают только функции продвижения пакетов по

готовой таблице маршрутизации. Такие маршрутизаторы являются усеченными маршрутизаторами, т. к. для их полноценной работы требуется наличие полнофункционального маршрутизатора, у которого можно взять готовую таблицу маршрутизации. Этот маршрутизатор часто называется сервером маршрутов. Отказ от самостоятельного выполнения функций построения таблицы маршрутизации резко удешевляет маршрутизатор и повышает его производительность.

Стирание граней между коммутаторами и маршрутизаторами

В классическом понимании коммутатор – это устройство, принимающее решение о продвижении пакетов на основании заголовков протоколов 2-го уровня, т. е. протоколов типа Ethernet или FDDI, а маршрутизатор – устройство, принимающее аналогичное решение на основании заголовков протоколов 3-го уровня, т. е. уровня протоколов IP или IPX. В настоящее время наблюдается отчетливая тенденция по совмещению в одном устройстве функций коммутатора и маршрутизатора.

Соотношение коммутации и маршрутизации в корпоративных сетях

До недавнего времени сложившимся информационным потокам корпоративной сети наилучшим образом соответствовала следующая иерархическая структура. На нижнем уровне (уровне отделов) располагались сегменты сети, построенные на быстро работающих повторителях и коммутаторах. Сегменты включали в себя как рабочие станции, так и серверы. В большинстве случаев было справедливо эмпирическое соотношение 80/20, в соответствии с которым основная часть трафика (80 %) циркулировала внутри сегмента, т. е. порождалась запросами пользователей рабочих станций к серверам своего же сегмента.

На более высоком уровне располагался маршрутизатор, к которому подключалось сравнительно небольшое количество внутренних сетей, построенных на коммутаторах. Через порты маршрутизатора проходил трафик обращений рабочих станций одних сетей к серверам других сетей. Известно, что маршрутизатор затрачивает больше времени на обработку каждого пакета, чем коммутатор, поскольку он выполняет более сложную обработку трафика, включая интеллектуальные алгоритмы фильтрации, выбор маршрута при наличии нескольких возможных путей и т. п. С другой стороны, трафик, проходящий через порты маршрутизатора, был менее интенсивный, чем внутрисегментный, поэтому сравнительно низкая производительность маршрутизатора не делала его узким местом.

Сегодня ситуация в корпоративных сетях быстро меняется. Количество пользователей стремительно растет. Пользователи избавляются от устаревающих текстовых приложений, отдавая предпочтение Web-интерфейсу. А завтра эти же пользователи будут работать с аудио-, видео-, push- и другими, абсолютно новыми приложениями, основанными на новых технологиях распространения пакетов, таких как IP Multicast и RSVP. Не работает и старое правило 80/20, сегодня большое количество информации берется из публичных серверов Internet, а также из Web-серверов других подразделений предприятия, создавая большой межсетевой трафик. Существующие сети не оптимизировались для таких непредсказуемых потоков трафика, когда каждый может общаться почти с каждым. А с проникновением в корпоративные сети технологии Gigabit Ethernet эта проблема обострится еще больше.

Таким образом, сегодня образовался большой разрыв между производительностью типичного маршрутизатора и типичного коммутатора. В этой ситуации возможны два решения: либо отказаться вообще от маршрутизации, либо увеличить ее производительность.

За последние годы основные усилия были сосредоточены в первом направлении – применять маршрутизацию как можно реже, только там, где от нее никак нельзя отказаться. Например, на границе между локальной и глобальной сетью. Отказ от маршрутизаторов означает переход к так называемой плоской сети, т. е. сети, построенной только на коммутаторах, а значит, и отказ от всех интеллектуальных возможностей обработки трафика, присущих маршрутизаторам. Такой подход повышает производительность, но приводит к потере всех преимуществ, которые давали маршрутизаторы, а именно:

– маршрутизаторы более надежно, чем коммутаторы, изолируют части большой составной сети друг от друга, защищая их от ошибочных кадров, порождаемых неисправным программным или аппаратным обеспечением других сетей (например, от широковещательных штормов);

– маршрутизаторы обладают более развитыми возможностями защиты от несанкционированного доступа за счет функций анализа и фильтрации трафика на более высоких уровнях – сетевом и транспортном;

– сеть, не разделенная маршрутизаторами, имеет ограничения на число узлов (для популярного протокола IP это ограничение составляет 254 узла для сетей самого доступного класса C).

Из этого следует, что в сети необходимо сохранять функции маршрутизации в привычном смысле этого слова.

Что касается второго направления – повышение производительности маршрутизаторов, – сложилось так, что самые активные действия в этом направлении были предприняты производителями коммутаторов, наделявшими свои продукты некоторыми возможностями маршрутизаторов. Именно в модифицированных коммутаторах были впервые достигнуты скорости маршрутизации в 5 – 7 миллионов пакетов в секунду, а также опробованы многие важные концепции ускорения функций маршрутизации.

Коммутаторы 3-го уровня с классической маршрутизацией

Термин «коммутатор 3-го уровня» употребляется для обозначения целого спектра коммутаторов различного типа, в которые встроены функции маршрутизации пакетов. Функции коммутации и маршрутизации могут быть совмещены двумя способами:

– классическим, когда маршрутизация выполняется по каждому пакету, требующему передачи из сети в сеть, а коммутация выполняется для пакетов, принадлежащих одной сети;

– нестандартным способом ускоренной маршрутизации, когда маршрутизируются несколько первых пакетов устойчивого потока, а все остальные пакеты этого потока коммутируются.

Классический коммутатор 3-го уровня подобно обычному коммутатору захватывает все кадры своими портами независимо от их MAC-адресов, а затем принимает решение о коммутации или маршрутизации каждого кадра. Если кадр имеет MAC-адрес назначения, отличный от MAC-адреса порта маршрутизатора, то этот кадр коммутируется. Если устройство поддерживает технику VLAN, то перед передачей кадра проверяется принадлежность адресов назначения и источника одной виртуальной сети. Если же кадр направлен непосредственно MAC-адресу какого-либо порта маршрутизатора, то он маршрутизируется стандартным образом. Коммутатор 3-го уровня может поддерживать динамические протоколы маршрутизации, такие как RIP или OSPF, а может полагаться на статическое задание маршрутов или на получение таблицы маршрутизации от другого маршрутизатора.

Такие комбинированные устройства появились сразу после разработки коммутаторов, поддерживающих виртуальные локальные сети (VLAN). Для Связи VLAN требовался маршрутизатор. Размещение маршрутизатора в одном корпусе с коммутатором позволяло получить некоторый выигрыш в производительности, например, за счет исключения одно-

го этапа буферизации пакета, когда он передается из коммутатора в маршрутизатор. Хотя такие устройства с равным успехом можно называть маршрутизирующими коммутаторами или коммутирующими маршрутизаторами, за ними закрепилось название коммутаторов 3-го уровня.

Более быстродействующей реализацией данного подхода являются устройства, в которых функции маршрутизации перенесены из универсального центрального процессора в специализированные микросхемы портов. При этом ускорение процесса маршрутизации происходит не только за счет распараллеливания работы между несколькими процессорами, но и за счет использования специализированных процессоров вместо универсальных.

Еще один тип коммутаторов 3-го уровня – это коммутаторы, которые ускоряют процесс маршрутизации за счет выявления устойчивых потоков в сети и обработки по схеме маршрутизации только нескольких первых пакетов потока.

Поток – это последовательность пакетов, имеющих некоторые общие свойства, по меньшей мере, у них должны совпадать адрес отправителя и адрес получателя, и тогда их можно отправлять по одному и тому же маршруту. Желательно, чтобы пакеты потока имели одно и то же требование к качеству обслуживания.

Если бы все коммутаторы/маршрутизаторы, изображенные на рис. 2.101, работали по классической схеме, то каждый пакет, отправляемый из рабочей станции, принадлежащей одной IP-сети, серверу, принадлежащему другой IP-сети, проходил бы через блоки маршрутизации всех трех устройств.

Рис. 2.101. Ускоренная маршрутизация потока пакетов

В схеме ускоренной маршрутизации такую обработку проходят только несколько первых пакетов долговременного потока, т. е. классическая схема работает до тех пор, пока долговременный поток не будет выявлен.

После этого первый коммутатор на пути следования потока выполняет нестандартную обработку пакета – он помещает в кадр канального протокола, например Ethernet, не MAC-адрес порта следующего маршрутизатора, а MAC-адрес узла назначения, который на рисунке обозначен как MAC. Как только эта замена произведена, кадр с таким MAC-адресом перестает поступать на блоки маршрутизации второго и третьего коммутатора/маршрутизатора, а проходит только через блоки коммутации этих устройств. Процесс передачи пакетов действительно ускоряется, т. к. они не проходят многократно повторяющиеся этапы маршрутизации. В то же время защитные свойства маршрутизаторы сохраняют, т. к. первые пакеты проверяются на допустимость передачи в сеть назначения, поэтому сохраняются фильтрация широковещательного шторма, защита от несанкционированного доступа и другие преимущества сети, разделенной на подсети.

Для реализации описанной схемы нужно решить несколько проблем. Первая – на основании каких признаков определяется долговременный поток. Это достаточно легкая проблема, и основные подходы к ее решению очевидны – совпадение адресов и портов соединения, общие признаки качества обслуживания, некоторый порог одинаковых пакетов для фиксации долговременного потока. Вторая проблема более серьезная: на основании какой информации первый маршрутизатор узнает MAC-адрес узла назначения. Этот узел находится за пределами непосредственно подключенных к первому маршрутизатору сетей, поэтому использование протокола ARP здесь не поможет. Именно здесь расходятся пути большинства фирменных технологий ускоренной маршрутизации. Многие компании разработали собственные служебные протоколы, с помощью которых маршрутизаторы запрашивают этот MAC-адрес друг у друга, пока последний на пути маршрутизатор не выяснит его с помощью протокола ARP.

Фирменные протоколы используют как распределенный подход, когда все маршрутизаторы равны в решении проблемы нахождения MAC-адреса, так и централизованный, когда в сети существует выделенный маршрутизатор, который помогает ее решить для всех.

Вопросы и задания для самопроверки

1. Всякое ли приложение, выполняемое в сети, можно назвать сетевым?
2. Что общего и в чем отличие между взаимодействием компьютеров в сети и взаимодействием компьютера с периферийным устройством?
3. Назовите главные недостатки полносвязной топологии, а также топологий типа «общая шина», «звезда», «кольцо».
4. В чем отличие логической структуризации сети от физической?
5. Что такое «открытая система»? Приведите примеры закрытых систем.
6. Что стандартизует модель OSI?
7. Дайте краткое описание функций каждого уровня и приведите примеры стандартных протоколов для каждого уровня модели OSI.
8. Назовите наиболее часто используемые характеристики производительности сети.
9. Что важнее для передачи мультимедийного трафика: надежность или синхронность?
10. Могут ли цифровые линии связи передавать аналоговые данные?
11. Назовите методы компрессии, наиболее подходящие для текстовой информации. Почему они неэффективны для сжатия двоичных данных?
12. Сеть с коммутацией пакетов испытывает перегрузку. Для устранения этой ситуации размер окна в протоколах компьютеров сети нужно увеличить или уменьшить?
13. Как влияет надежность линий связи в сети на выбор размера окна?
14. В чем проявляется избыточность TDM-технологии?
15. Какой способ коммутации более эффективен для передачи пульсирующего трафика: коммутация каналов или коммутация пакетов?
16. Поясните разницу между расширяемостью и масштабируемостью на примере технологии Ethernet.
17. В чем заключается суть протокола CSMA/CD?
18. Что такое коллизия?
19. Что такое домен коллизий?
20. Чем объясняется то, что минимальный размер кадра в стандарте 10Base-5 был выбран равным 64 байт?
21. При каких типах ошибок в сети Ethernet концентратор обычно отключает порт?

22. Из каких соображений выбрана максимальная длина физического сегмента в стандартах Ethernet?

23. Опишите алгоритм доступа к среде технологии Token Ring.

24. В чем состоит сходство и различие технологий FDDI и Token Ring?

25. С чем связано ограничение, известное как «правило 4-х хабов»?

26. Что такое структурированная кабельная система?

27. Что означает термин *backbone*?

28. Каким образом мост/коммутатор строит свою внутреннюю таблицу?

29. Можно ли утверждать, что у любого моста скорость продвижения не выше скорости фильтрации?

30. Что произойдет, если в сети, построенной на концентраторах, имеются замкнутые контуры?

31. Почему полнодуплексный Ethernet не поддерживается в концентраторах?

32. Каким образом коммутатор может управлять потоком пакетов, поступающих от сетевых адаптеров станций сети?

33. Какая информация содержится в таблицах мостов/коммутаторов и маршрутизаторов?

34. Пусть IP-адрес некоторого узла подсети равен 198.65.12.67, а значение маски для этой подсети – 255.255.255.240. Определите номер подсети. Какое максимальное число узлов может быть в этой подсети?

35. Какие метрики расстояния могут быть использованы в алгоритмах сбора маршрутной информации?

36. Каким образом должен быть сконфигурирован маршрутизатор, чтобы он предотвращал «широковещательный шторм»?

37. За счет чего коммутаторы третьего уровня ускоряют процесс маршрутизации?

ЛИТЕРАТУРА

1. Шпаковский, Г.И. Организация параллельных ЭВМ и суперскалярных процессоров: учеб. пособие для вузов. – М.: Университетское, 1996.

2. Олифер, В.Г. Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов. – 4-е изд. / В.Г. Олифер, Н.А. Олифер. – Спб.: Питер, 2010.

3. Столингс, В. Современные компьютерные сети / В. Столлингс. – СПб: Питер, 2003.

4. Орлов, С. Организация ЭВМ и систем / С. Орлов, Б. Цилькер. – СПб: Питер, 2004.

5. Андэрсон, К. Локальные сети / К. Андэрсон, М. Минаси. – СПб.: Корона, 1999.

6. Хелд, Г. Технологии передачи данных / Г. Хелд. – СПб: Питер, 2003.

7. Семенов, А. Проектирование и расчет структурированных кабельных систем и их компонентов / А. Семенов. – СПб.: ДМК, 2003.

8. Кульгин, М.В. Компьютерные сети. Практика построения / М.В. Кульгин. – СПб., 2003.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ ... 3
МОДУЛЬ 1. ВЫЧИСЛИТЕЛЬНЫЕ КОМПЛЕКСЫ И СИСТЕМЫ ... 8
1.1. Принципы параллельной обработки информации ... 8
1.2. Многомашинные вычислительные комплексы ... 12
1.3. Многопроцессорные вычислительные комплексы ... 17
1.4. Конвейерные вычислительные системы ... 25
1.5. Матричные вычислительные системы ... 32
1.6. Ассоциативные и систолические системы ... 41
1.7. Функционально распределенные и перестраиваемые вычислительные системы 54
1.8. Вопросы и задания для самопроверки ... 65
МОДУЛЬ 2. Компьютерные сети ... 66
2.1. Эволюция вычислительных систем. Основные программные и аппаратные компоненты сети ... 67
2.2. Понятие «Открытая система». Модель OSI ... 74
2.3. Стандартные стеки коммуникационных протоколов ... 86
2.4. Линии связи. Типы, аппаратура, характеристики линий связи. ... 92
2.5. Стандарты кабелей ... 98
2.6. Методы передачи дискретных данных на физическом уровне ... 106
2.7. Методы передачи данных канального уровня ... 116
2.8. Методы коммутации: коммутация каналов и коммутация пакетов ... 130
2.9. Протоколы и стандарты локальных сетей. Структура стандартов IEEE 802.X . 146
2.10. Протокол LLC уровня управления логическим каналом ... 153
2.11. Технология Ethernet (802.3). Метод доступа CSMA/CD ... 159
2.12. Спецификации физической среды Ethernet. Стандарт 10Base-5 ... 168
2.13. Технология Ethernet. Стандарты 10Base-2, 10Base-T, 10Base-F. Понятие домена коллизий ... 172
2.14. Технология Token Ring (802.5) ... 179
2.15. Технология FDDI ... 185
2.16. Технология Fast Ethernet ... 193
2.17. Высокоскоростная технология Gigabit Ethernet ... 203
2.18. Структурированная кабельная система ... 209
2.19. Сетевые адаптеры и концентраторы ... 216
2.20. Логическая структуризация сети с помощью мостов и коммутаторов ... 228
2.21. Принципы работы мостов ... 236
2.22. Коммутаторы локальных сетей ... 245
2.23. Техническая реализация коммутаторов ... 255
2.24. Характеристики, влияющие на производительность коммутаторов, дополнительные функции ... 262
2.25. Виртуальные локальные сети. Типовые схемы применения коммутаторов в локальных сетях ... 273
2.26. Принципы объединения сетей на основе протоколов сетевого уровня ... 283
2.27. Адресация в IP-сетях ... 292
2.28. Протокол IP ... 310
2.29. Протоколы маршрутизации в IP-сетях ... 315
2.30. Основные характеристики маршрутизаторов. Стирание граней между маршрутизаторами и коммутаторами ... 332
2.31. Вопросы и задания для самопроверки ... 344
Литература ... 346

Учебное издание

РУГОЛЬ Дмитрий Геннадьевич

ВЫЧИСЛИТЕЛЬНЫЕ КОМПЛЕКСЫ, СИСТЕМЫ И СЕТИ

Учебно-методический комплекс
для студентов специальностей 1-40 02 01 «Вычислительные машины, системы и сети», 1-40 01 01 «Программное обеспечение информационных технологий»

Редактор *Т. В. Булах*

Дизайн обложки *А. Н. Парфёновой*

Подписано в печать 30.06.2014. Формат 60×84 1/16. Бумага офсетная. Ризография
Усл. печ. л. 20,88. Уч.-изд. л. 19,7. Тираж 30 экз. Заказ 960.

Издатель и полиграфическое исполнение –
учреждение образования «Полоцкий государственный университет».

Свидетельство о государственной регистрации
издателя, изготовителя, распространителя печатных изданий
№1/305 от 22.04.2014.

ЛП № 02330/494255 от 08.05.2014.

Ул. Блохина, 29, 211440, г. Новополоцк.