ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ.

ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ (ЧАСТЬ 1)

ГУПК
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПРОМЫШЛЕННЫХ КОММУНИКАЦИЙ

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ - УЧЕБНО-НАУЧНО-
ПРОИЗВОДСТВЕННЫЙ КОМПЛЕКС»

В.Т. Еременко, В.А. Лобанова, А.В. Тютякин,
В.М. Донцов, А.Е. Георгиевский, О.А. Воронина

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ.

ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ (ЧАСТЬ 1)

Рекомендовано ФГБОУ ВПО «Госуниверситет - УНПК»
для использования в учебном процессе в качестве конспекта лекций
для высшего профессионального образования

Орел 2012

УДК 004.72 (075)
ББК 32.973. 02я7
В94

Рецензенты:

доктор технических наук, профессор кафедры
«Электроника, вычислительная техника и информационная безопасность»
Федерального государственного бюджетного образовательного
учреждения высшего профессионального образования
«Государственный университет - учебно-научно-
производственный комплекс»
А.И. Суздальцев,

доктор технических наук, профессор кафедры № 7
Академии Федеральной службы охраны Российской Федерации
Б.Р. Иванов

В94 **Вычислительная техника и информационные технологии. Вычислительные сети. Ч. 1:** конспект лекций для высшего профессионального образования / В.Т. Еременко [и др.]. – Орел: ФГБОУ ВПО «Госуниверситет - УНПК», 2012. – 334 с.

ISBN 978-5-93932-461-8

Часть 1 конспекта лекций посвящена общим вопросам реализации вычислительных сетей, а также коммуникационным сетевым технологиям физического, канального и сетевого уровней. Часть 2 настоящего конспекта планируется посвятить сетевым технологиям «сквозных» уровней (транспортного, сеансового, представительского и прикладного).

Предназначен студентам, обучающимся по направлению 210700.62 «Инфокоммуникационные технологии и системы связи», при изучении дисциплины «Вычислительная техника и информационные технологии», а также может быть полезен студентам, обучающимся по другим направлениям в области информационных технологий.

УДК 004.72 (075)
ББК 32.973. 02я7

ISBN 978-5-93932-461-8

СОДЕРЖАНИЕ

СОКРАЩЕНИЯ

АС – автоматизированная система

АСУ – автоматизированная система управления

ВС – вычислительная сеть

ЗИ – защита информации

ИБ – информационная безопасность

ИО – информационный обмен

КИ – ключевая информация

КС – канал связи

ЛВС – локальная вычислительная сеть

НСД – несанкционированный доступ

ОС – операционная система

ПБ – политика безопасности

ПК – персональный компьютер

ПО – программное обеспечение

ЦП – цифровая подпись

ЦФК – центр формирования ключей

ЭВМ – электронно-вычислительная машина

1. ОБЩИЕ ВОПРОСЫ РЕАЛИЗАЦИИ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ

1.1. Определение, классификация и основные параметры вычислительной сети

В настоящее время вычислительные сети (ВС) являются одной из главных составляющих информационных технологий и одной из наиболее важных технологий с точки зрения влияния на человеческую деятельность в целом.

1.1.1. Определение ВС

Наиболее общим и полным *определением ВС*, с точки зрения авторов, является следующее [3, 7].

Вычислительная сеть есть совокупность объединенных некоторой системой связи территориально распределенных компьютеров и/или терминалов, ориентированная на коллективное использование указанными компьютерами/терминалами общесетевых аппаратных, программных и информационных ресурсов.

Вкратце прокомментируем вышеприведенное определение.

Итак, *основными абонентами* ВС являются компьютеры, в общем случае – различного класса: персональные компьютеры, серверы, большие компьютеры, реже – суперкомпьютеры. В качестве абонентов ВС также могут выступать терминалы. Под ними, в первую очередь, подразумеваются средства интерактивного взаимодействия пользователей с многопользовательским компьютером; «классический» вариант терминала – монитор с клавиатурой. Следует отметить, что каждый из компьютеров ВС работает автономно, под управлением своей операционной системы (ОС), при отсутствии общей ОС в подавляющем большинстве практических случаев, а взаимодействие между компьютерами ВС сводится только к обмену сообщениями. Поэтому ВС относят к *слабосвязанным* вычислительным системам [7].

Важным признаком ВС является ее *территориально распределенный характер*: абоненты ВС обычно находятся на существенном расстоянии друг от друга, как минимум, на порядок превышающем типовое расстояние между узлами и блоками пространственно сосредото-

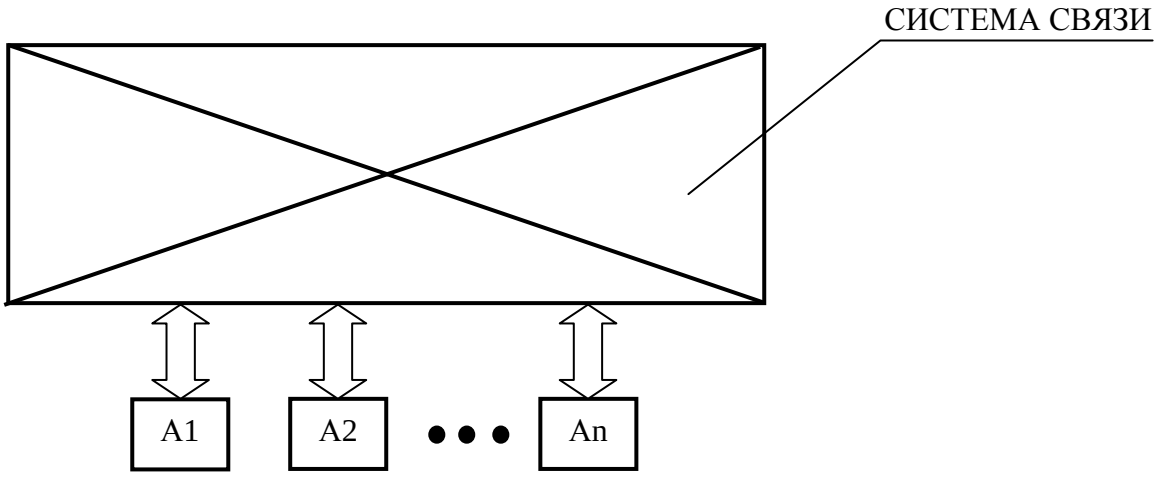
ченных вычислительных систем. Конкретные значения расстояния между абонентами зависят от типа и назначения ВС и, в общем, могут находиться в пределах от нескольких метров (локальные ВС подразделений предприятий и организаций) до тысяч километров (глобальные ВС мирового уровня, в первую очередь, – Интернет). Таким образом, ВС являются *распределенными вычислительными системами* [7].

Основным назначением ВС, как следует из вышеприведенного определения, является обеспечение доступа ее абонентов к некоторым общесетевым ресурсам. В их качестве могут выступать:

- сетевые устройства, например, сетевой принтер, доступный для использования всеми персональными компьютерами некоторой офисной локальной ВС;
- программное обеспечение, например, программа формирования финансового отчета организации, доступная для ввода и корректировки данных всем ее подразделениям;
- информационные ресурсы, например, представленные на Internet-сайте некоторой фирмы каталог и технические описания выпускаемой продукции в электронном виде, доступные всем пользователям Интернет.

Для обеспечения доступа абонентов ВС к общесетевым ресурсам, очевидно, все они должны быть объединены между собой некоторой *системой связи*. Данная система представляет собой совокупность аппаратных и программных средств, обеспечивающих обмен данными между абонентами ВС. Степень сложности и структура данной системы зависят от типа и назначения ВС. Так, система связи простейших (односегментных) локальных ВС может включать в себя, по существу, только сетевые адаптеры абонентов и соединительный кабель. С другой стороны, системы связи сложных локальных и, тем более, глобальных ВС являются весьма сложными и многоуровневыми. Очевидно, именно система связи ВС определяет эффективность обмена данными между абонентами ВС и, как следствие, эффективность работы ВС в целом, являясь при этом *основным компонентом ВС*. Поэтому основной проблемой теории и практики ВС является создание протоколов, алгоритмов, аппаратных и программных средств, обеспечивающих максимально эффективный обмен данными между абонентами какой-либо разновидности ВС или конкретной ВС. В первую очередь, рассмотрению вышеуказанных протоколов, алгоритмов и средств посвящается данный учебник.

Исходя из вышесказанного, *обобщенная структурная схема ВС* имеет вид, представленный на рис. 1.1.



A1,…, An – абоненты ВС

Рис. 1.1. Обобщенная структурная схема ВС

Рис. 1.1 не нуждается в особых комментариях. Необходимо только отметить, что, как указано ранее, абоненты и система связи пространственно распределены, а состав и структура данной системы определяются типом и назначением ВС. Поэтому практически невозможно представить систему связи в виде универсальной структурной схемы, более подробной, чем на рис. 1.1. Подобное представление возможно только для каждого из конкретных типов ВС, состав и структура систем связи которых будут рассматриваться в соответствующих разделах (см. далее).

Исходя из вышесказанного, использование ВС предоставляет ее пользователям следующие *основные преимущества* [2, 3]:

- быстрый доступ непосредственно с рабочего места к источникам информации различного уровня (от корпоративного до глобального) и характера (научным, образовательным и справочным материалам, новостям и т. п.), что весьма существенно повышает эффективность труда практически во всех областях человеческой деятельности;

- оперативный обмен сообщениями и документацией, быстрые переводы денежных средств между организациями и физическими лицами за счет возможности применения таких технологий, как электронная почта, IP-телефония, on-line-совещания и конференции, «киберденьги» и т. п., что также значительно повышает эффективность работы, в том числе обеспечивает быстрое и качественное принятие решений;

- возможность разделения пользователями дорогостоящих аппаратно-программных общесетевых ресурсов, что существенно снижает как общие материальные затраты на создание ВС, так и индивидуальные затраты каждого из пользователей;

- свобода в территориальном размещении компьютеров, т. е. возможность формирования информационной системы, расположение абонентов которой определяется только требованиями к эффективности работы и удобству пользователей, но не какими-либо принципиальными техническими ограничениями.

Благодаря вышеперечисленным преимуществам, ВС в настоящее время, как указано ранее, являются одной из важнейших компонент информационных технологий, в значительной степени определяя технологический облик современного общества.

1.1.2. Классификация ВС

Существуют различные признаки классификации ВС [2, 3, 7]. Однако наиболее распространена классификация по *территориальному* признаку, т. е. по размерам территории, на которой распределены абоненты ВС. Данный признак классификации выбран в качестве наиболее важного, поскольку протоколы, алгоритмы и средства обмена данными между абонентами ВС определяются, в первую очередь, именно размерами территории, «покрываемой» ВС.

По территориальному признаку выделяют следующие основные типы ВС:

- *локальные ВС* (ЛВС), в англоязычной литературе – Local Area Networks (LAN);

- *глобальные ВС* (ГВС), по-англ. – Wide Area Networks (WAN).

К *ЛВС* относят сети с относительно небольшим числом абонентов (обычно порядка нескольких десятков – сотен), распределенных по относительно небольшой территории (не более нескольких километров в радиусе). Благодаря этому в системах связи ЛВС имеется возможность применения сравнительно дорогих высококачественных линий связи, выделяемых специально для ЛВС и позволяющих достигать высоких скоростей обмена данными (порядка сотен мегабит – единиц гигабит в секунду). Соответственно, протоколы, алгоритмы и средства обмена данными в ЛВС ориентированы на высокоскорост-

ной обмен по высококачественным линиям связи между относительно малым числом абонентов, распределенных по сравнительно небольшой территории.

ГВС объединяют большое, по сравнению с ЛВС, число абонентов (до десятков – сотен миллионов), распределенных по значительной территории, часто находящихся в различных городах и странах. Обычно группы абонентов ГВС бывают объединены между собой в ЛВС, причем часто данные ЛВС *разнородны* с точки зрения используемых стандартов, протоколов и средств обмена данными между абонентами ЛВС. Поэтому в большинстве практических случаев ГВС представляет собой *составную сеть*, т. е. совокупность территориально распределенных и, в общем случае, разнородных ЛВС, объединенных между собой системой связи. При этом, ввиду технической и организационно-экономической сложности прокладки высококачественных линий связи на большие расстояния, в качестве элементов системы связи ГВС, наряду со специально выделенными, часто применяются каналы связи общего пользования, например, телефонные. Это предполагает, например, более сложные процедуры контроля и восстановления данных, чем в ЛВС. Вследствие этого протоколы, алгоритмы и средства обмена данными в ГВС ориентированы на организацию связи между абонентами составной сети, территориально распределенными по различным городам, странам и континентам, с использованием как специально выделенных каналов связи, так и каналов связи общего пользования, при отсутствии гарантии высокого качества обмена данными.

Необходимо отметить, что по территориальному признаку, кроме ЛВС и ГВС, выделяют также ряд промежуточных типов ВС [3, 7], в частности, *городские ВС*, или сети мегаполисов (в англоязычной литературе – Metropolitan Area Networks, MAN). Они предназначены для обслуживания территорий крупных городов – мегаполисов, и сочетают в себе ряд характерных особенностей как ЛВС, так и ГВС. Их рассмотрение выходит за рамки настоящего учебного пособия; интересующиеся лица могут ознакомиться с особенностями реализации городских ВС, например, по литературным источникам [3] и [7].

Следует также отметить, что в настоящее время наблюдается тенденция к сближению принципов реализации ЛВС и ГВС [1]. Подробно эти вопросы будут рассматриваться по мере изложения материалов данного учебного пособия (см. далее).

Кроме пространственного, известны и другие критерии классификации ВС, например, *масштаб подразделения (организации)*, в пределах которого действует ВС. Различают ВС отделов, кампусов (университетских городков) и корпоративные (т. е. ВС крупных фирм, предприятий или организаций, называемые по-англ. Enterprise-Wide Networks) [1]. В целом, данные типы ВС являются частными случаями ЛВС (корпоративные ВС в ряде случаев также являются разновидностями ГВС). При этом принципы и средства их реализации существенно не отличаются от принципов и средств реализации других разновидностей ЛВС или, соответственно, ГВС, примерно аналогичных им по числу абонентов и площади их размещения. Поэтому при дальнейшем изложении материала масштаб подразделения, «покрываемого» сетью, не будет использоваться в качестве базового критерия классификации ВС, а ВС отделов и кампусов, а также корпоративные ВС будут рассматриваться только как показательные примеры реализации ЛВС или ГВС.

Основные характеристики ВС

К ним относятся следующие [3]:

- производительность;
- надежность;
- информационная безопасность;
- расширяемость;
- масштабируемость;
- прозрачность;
- виды трафика (т. е. потоков данных), поддерживаемые сетью;
- управляемость;
- совместимость (интегрируемость).

Производительность ВС характеризуют следующие основные параметры [3]:

- время реакции;
- пропускная способность;
- задержка передачи.

Время реакции определяется как интервал времени между подачей пользователем запроса к какой-либо сетевой службе и получением ответа на этот запрос. Оно является одним из базовых *пользовательских* параметров ВС.

Время реакции ВС обычно складывается из нескольких составляющих. В общем случае в него входит время подготовки запросов на

пользовательском компьютере, время передачи запросов между пользователем и сервером через сегменты сети и промежуточное коммуникационное оборудование, время обработки запросов на сервере, время передачи ответов от сервера пользователю и время обработки получаемых от сервера ответов на пользовательском компьютере. Очевидно, численное значение данного параметра зависит от многих факторов, в том числе от типа сетевой службы, к которой обращается пользователь, степени загруженности линий связи и коммуникационного оборудования и т. п. Поэтому на практике используют также средневзвешенную оценку времени реакции сети, усредняя этот показатель по пользователям, сегментам ВС и времени суток.

Пропускная способность ВС определяется как объем данных, передаваемых между некоторыми двумя узлами ВС или между всеми ее узлами (общая пропускная способность), в единицу времени. В отличие от времени реакции, она не является пользовательским параметром ВС, а характеризует качество выполнения основной «внутренней» функции ВС – передачи информации, и поэтому чаще используется при анализе производительности ВС, чем время реакции. Пропускная способность обычно выражается в единицах информации (битах, байтах, килобайтах и т. п.) в секунду. Различают среднюю, мгновенную и максимальную пропускную способность ВС.

Средняя пропускная способность вычисляется как отношение общего объема данных, переданных за некоторый интервал времени, к его длительности. При этом последняя выбирается достаточно большой (не менее часа, иногда сутки или неделя).

Мгновенная пропускная способность определяется аналогично средней, за исключением того, что длительность интервала передачи выбирается достаточно малой (порядка десятков – сотен миллисекунд).

Максимальная пропускная способность определяется как наибольшая мгновенная пропускная способность, зафиксированная в течение некоторого периода наблюдения.

Наиболее информативными являются такие параметры ВС, как средняя и максимальная пропускные способности [3]. Средняя пропускная способность характеризует производительность ВС на достаточно длительном интервале времени, в течение которого в соответствии с законами статистики максимумы и минимумы интенсивности обмена сообщениями между абонентами ВС компенсируют друг друга. Максимальная пропускная способность характеризует возможно-

сти ВС справляться с пиковыми нагрузками, характерными для периодов ее наиболее интенсивной работы, например, для начала рабочего дня в некоторой организации, когда ее сотрудники практически одновременно обращаются к общесетевым информационным ресурсам (например, базам данных) [3].

Задержка передачи определяется как длительность интервала времени между моментом поступления данных на вход какого-либо сетевого устройства или фрагмента сети и моментом их появления на выходе этого устройства (фрагмента) [3]. Этот параметр отличается от времени реакции тем, что характеризует только систему связи ВС или ее отдельные фрагменты, без задержек, вносимых пользовательскими компьютерами ВС. Численно задержка передачи описывается двумя величинами: максимальной задержкой передачи и вариацией (т. е. пределами изменения) задержки. Типовые значения максимальной задержки передачи составляют порядка сотен миллисекунд – единиц секунд [3]. Задержка передачи существенно влияет на качество обмена аудио- и видеоинформацией в реальном масштабе времени между абонентами ВС (например, в IP-телефонии, при проведении on-line-совещаний, и т. п.). Однако данный параметр сравнительно маловажен с точки зрения пользователя, например, при обмене сообщениями между абонентами ВС по электронной почте, при передаче/приеме текстовой и графической документации в электронном виде и т. п.

В заключение следует отметить, что, несмотря на кажущуюся взаимную зависимость пропускной способности и задержки передачи, реально они являются независимыми параметрами. В частности, на практике возможны ситуации, при которых некоторый фрагмент ВС обладает высокой пропускной способностью при большом времени задержки, и наоборот (см., например, [3]).

Надежность ВС характеризуется следующими основными параметрами [3]:

- коэффициентом готовности;
- вероятностью доставки без искажений некоторого фрагмента данных фиксированной длины (пакета) в узел назначения;
- отказоустойчивостью.

Коэффициент готовности ВС определяется как отношение суммарной длительности интервалов времени, в течение которых ВС может реально использоваться абонентами, к суммарному времени эксплуатации ВС (включающему в себя и время простоев, восстановления после отказов и т. п.). Коэффициент готовности является базовой количественной характеристикой надежности ВС.

Вероятность доставки пакета без искажений равна отношению количества пакетов, доставленных без искажения в некоторый узел ВС, к общему количеству переданных в него пакетов. Естественно, в общем случае значение данного параметра может быть различным для различных узлов и фрагментов ВС. Используются и другие, взаимосвязанные с ним параметры: вероятность потери пакета, отношение числа потерянных пакетов к числу доставленных и др. [3].

В отличие от двух вышеперечисленных количественных параметров ВС, ее *отказоустойчивость* является качественной характеристикой надежности ВС. Под отказоустойчивостью ВС понимается ее способность скрыть от пользователя отказ ее отдельных компонентов (блоков, узлов или фрагментов). Например, если копии базы данных хранятся одновременно на нескольких файловых серверах, то отказ одного из них будет практически незаметен для пользователей ВС [3].

Информационная безопасность является качественной характеристикой ВС, отражающей способность ВС к защите данных от несанкционированного доступа. На данную характеристику влияют многие факторы, из которых основными являются [1, 3]:

- методы кодирования данных, применяемые абонентами ВС;
- степень территориального распределения абонентов ВС и ее структура;
- типы используемых линий связи (проводные или беспроводные, специально выделенные или общего пользования, например, телефонные, и т. п.).

Следует отметить, что вопросы информационной безопасности ВС рассматриваются в рамках специальных дисциплин, изучаемых в процессе подготовки бакалавров по направлению «Инфокоммуникационные технологии и системы связи». Поэтому в настоящем учебном пособии данные вопросы отражаться не будут.

Расширяемость и *масштабируемость* ВС являются формально близкими по смыслу, но все же самостоятельными характеристиками ВС.

Расширяемость ВС характеризует возможность технически и организационно простой и существенно не влияющей на производительность ВС реализации следующих процедур:

- введения в ее состав дополнительных абонентов;
- наращивания длины линий связи;
- обновления аппаратных средств ВС.

Масштабируемость ВС, в свою очередь, определяет возможность увеличения количества абонентов ВС и наращивания охватываемой ею территории в широких пределах, также без существенного влияния на производительность ВС. В целом, хорошей масштабируемостью обладает многосегментная сеть, имеющая иерархическую структуру и построенная с использованием специального коммуникационного оборудования – коммутаторов и маршрутизаторов (общие принципы реализации и применения которых изложены в гл. 3 и 4) [3, 7].

При этом ВС может обладать, например, хорошей расширяемостью при ограниченной масштабируемости. Примером такой ВС является ЛВС, включающая один сегмент на основе коммутатора (п. 3.7), которая позволяет без существенных технических и организационных сложностей подключать до нескольких десятков абонентов (компьютеров) [3]. Однако подключение большего числа абонентов потребует применения дополнительного коммуникационного оборудования и структуризации ЛВС (например, разбиения ее на несколько сегментов). В принципе, возможны и варианты ВС, обладающие ограниченной расширяемостью при хорошей масштабируемости [3, 6].

В целом, расширяемость и масштабируемость ВС являются ее качественными характеристиками. Однако частично они могут быть описаны и посредством ряда количественных параметров, например, максимальным числом абонентов, которое может быть подключено к ВС при ее расширении и т. п.

Прозрачность ВС характеризует ее свойство восприниматься пользователем не как множество отдельных компьютеров, связанных между собой системой связи, а как единый компьютер [3]. Различают прозрачность *на уровне пользователя* и *на уровне программиста* [3]. *Прозрачность на уровне пользователя* означает, что для работы с удаленными ресурсами он использует те же команды и процедуры, что и для работы с локальными ресурсами. *На уровне программиста* прозрачность заключается в том, что приложению для доступа к удаленным ресурсам требуются те же вызовы, что и для доступа к локальным ресурсам.

Концепция прозрачности может быть применена к различным аспектам работы ВС [3]. Например, *прозрачность расположения* означает, что от пользователя не требуется знаний о месте расположения программных и аппаратных ресурсов, таких как процессоры, принтеры, файлы и базы данных. *Прозрачность перемещения* означает, что

ресурсы должны свободно перемещаться из одного компьютера в другой без изменения их имен. *Прозрачность параллелизма* заключается в том, что процесс распараллеливания вычислений по процессорам и компьютерам ВС происходит автоматически, без участия программиста.

В настоящее время свойство прозрачности в полной мере не присуще многим из ВС, это скорее цель, к которой необходимо стремиться разработчикам современных ВС [3].

Виды трафика, поддерживаемые ВС. Данная характеристика отражает виды потоков данных, обмен которыми обеспечивает соответствующая ВС. Вообще говоря, существует два основных вида потоков данных (трафика) ВС [3, 7]:

- компьютерный трафик;
- мультимедийный трафик.

Первый из них представляет собой поток сообщений, содержащих, например, текстовые или графические документы в электронном виде (или их фрагменты), программное обеспечение и т. п., т. е. «традиционные» компьютерные данные. Мультимедийный же трафик предполагает, в первую очередь, обмен аудио- и видеоинформацией в реальном масштабе времени между абонентами ВС (что имеет место, например, в IP-телефонии, при проведении on-line-совещаний и конференций и т. п.).

Указанные виды трафика предъявляют принципиально различные требования к параметрам и характеристикам ВС. В частности, компьютерный трафик требует высокой надежности доставки сообщений получателю без ошибок (так как, например, часто является недопустимым искажение даже одного символа текстового документа или программы). С другой стороны, компьютерный трафик не предполагает обмена данными в реальном масштабе времени, поэтому он не предъявляет жестких требований к производительности ВС (см. выше). Мультимедийный же трафик, напротив, требует высокой пропускной способности ВС и малых задержек передачи ввиду необходимости обмена сообщениями в реальном масштабе времени. Однако допустим некоторый уровень искажений мультимедийных данных (определяемый требованиями к качеству передачи). Поэтому мультимедийный трафик не предъявляет столь жестких требований к надежности доставки сообщений, как компьютерный.

Исходя из вышеуказанных различий, протоколы, алгоритмы и средства обмена данными при компьютерном и мультимедийном характере трафика также существенно различаются. В принципе, не представляет значительной трудности реализация ВС с каким-либо одним видом трафика. Более сложным является построение ВС, совмещающих оба названных вида трафика. В таких ВС мультимедийный трафик часто относят к факультативному [3], поэтому его качеством при этом «жертвуют» в пользу качества компьютерного трафика. Однако в настоящее время активно ведутся работы и достигнуты определенные успехи по созданию протоколов, алгоритмов и средств обмена данными, обеспечивающих одинаково высокое качество как компьютерного, так и мультимедийного трафика [3].

Управляемость ВС характеризует возможность централизованно контролировать состояние ее основных элементов, выявлять и разрешать проблемы, возникающие при работе ВС, выполнять анализ ее производительности и планировать развитие. В идеале средства управления ВС представляют собой систему, осуществляющую наблюдение, контроль и управление каждым компонентом сети – от простейших до самых сложных устройств, при этом такая система рассматривает сеть как единое целое, а не как разрозненный набор отдельных устройств [3, 8]. Качественная система управления ВС осуществляет постоянное наблюдение за ней и, обнаружив какую-либо проблему, активизирует определенные действия, устраняющие ее, а также уведомляет сетевого администратора о том, что произошло и какие действия предприняты. Одновременно с этим система управления должна накапливать данные, на основании которых можно планировать развитие ВС. Кроме того, система управления должна обладать удобным интерфейсом, позволяющим выполнять все действия с одной консоли [3, 8].

Создание систем управления ВС, обладающих вышеперечисленными свойствами, является в значительной степени нерешенной проблемой. Однако, работы в данной области ведутся достаточно активно [3, 8].

Совместимость, или *интегрируемость* ВС характеризует ее способность использовать разнообразное программное и аппаратное обеспечение, в том числе различные операционные системы, разнородные аппаратные и программные средства и приложения от разных производителей и т. п. Такие ВС называются интегрированными [3].

Основной путь построения интегрированных сетей – использование модулей, выполненных в соответствии с *открытыми* стандартами и спецификациями. Данным вопросам посвящен п. 1.3.

1.2. Основные проблемы реализации ВС

Исходя из вышесказанного, при построении ВС практически любого типа и назначения, для обеспечения приемлемых параметров и характеристик ВС (см. ранее) необходимо решать следующие основные проблемы [1, 3, 7]:

- организация надежного и высокопроизводительного обмена данными между абонентами ВС по линиям связи (общего пользования или/и выделенным, индивидуальным или разделяемым и т. п.), в том числе кодирования и декодирования информации, проверки правильности передачи/приема данных и т. п.;
- выбор рациональных, с точки зрения производительности, надежности, расширяемости и масштабируемости, структуры ВС, конфигурации (топологии) физических связей между ее абонентами (т. е. электрических соединений между ними), а также логических связей между абонентами ВС (т. е. маршрутов передачи данных);
- создание системы иерархической адресации абонентов ВС (подобной системе почтовых адресов), обеспечивающей уникальную идентификацию абонента в ВС любого типа и масштаба и сводящей к минимуму ручной труд при назначении адресов и вероятность их дублирования;
- обеспечение совместной работы в составе ВС разнотипных и разнородных аппаратных и программных средств;
- организация «прозрачного» (т. е. независимого с точки зрения пользователя от состава, физической и логической топологии ВС) доступа абонентов ВС к общесетевым ресурсам (информационным, аппаратным или программным).

1.2.1. Организация обмена данными между абонентами ВС

Для реализации обмена данными по системе связи ВС необходимо решение следующих основных задач [1, 3, 7]:

- кодирование подлежащих передаче данных, обеспечивающее их максимально компактное представление, устойчивость процесса обмена данными к их искажениям в каналах связи, а также, при необходимости, защиту данных от несанкционированного доступа;

- оформление подвергнутых кодированию данных в виде единиц информации, формат которых обеспечивает их доставку абоненту, которому они предназначены, и их корректное распознавание им;

- представление данных сигналами-носителями с параметрами и характеристиками, предпочтительными для передачи по тому или иному типу канала связи;

- извлечение (детектирование) данных, представляемых сигналом-носителем, из этого сигнала, их корректное распознавание и декодирование на приемной стороне;

- установление и разрыв соединения между абонентами;

- коммутация и маршрутизация данных, т. е. обеспечение их эффективной передачи от абонента-источника к абоненту-приемнику через цепь промежуточных коммутационных узлов системы связи ВС.

Кодирование данных состоит в преобразовании исходной, подлежащей передаче двоичной последовательности, в другую последовательность, которая, в общем случае, должна удовлетворять следующим требованиям:

- отсутствие избыточности, т. е. минимально необходимый для восстановления исходных данных объем сообщения;

- возможность обнаружения и исправления ошибок в сообщении, вызванных искажениями сигнала-носителя данных в канале связи;

- защита передаваемых данных от несанкционированного доступа, т. е. возможность их восстановления только при известных на приемной стороне параметрах алгоритма декодирования.

Процедуры кодирования, обеспечивающие удовлетворение вышеперечисленных требований, известны под названиями *сжатия данных*, *помехоустойчивого кодирования* и *криптографического кодирования*. Вопросы реализации этих процедур освещены в гл. 3 и 4.

Оформление подлежащих передаче данных состоит в их представлении в виде двоичной последовательности определенного формата, в простейшем случае, содержащей:

- заранее оговоренные коды (идентификаторы) начала и конца последовательности;

- адреса отправителя и получателя;

- код типа последовательности (пользовательские данные, запрос соединения, запрос разъединения, запрос повторной передачи сообщения и т. п.);

- поле собственно данных;

- контрольные разряды, используемые для проверки наличия ошибок в последовательности и (в ряде случаев) их исправления.

Типы, структуры и форматы последовательностей, применяемых для обмена данными между абонентами ВС, вкратце будут рассмотрены далее, в подп. 1.2.2 (рис. 1.6 и пояснения к нему). Более подробно они будут освещены в гл. 3 и 4.

Представление подлежащих передаче данных сигналами-носителями обеспечивает собственно физическую передачу информации по каналам связи ВС. Носителями данных в указанных каналах служат напряжения, токи или электромагнитное излучение микроволнового или оптического диапазона, какие-либо параметры которых (амплитуда, частота, фаза и т. п.) являются функциями от передаваемой двоичной последовательности. На первый взгляд, в качестве сигнала-носителя рационально было бы использовать непосредственно передаваемую последовательность нулей и единиц, представляемых определенными уровнями напряжения, тока или излучения. Однако по ряду причин, подробно описанных в п. 2.4, в большинстве практических случаев такое представление данных в каналах связи ВС или не обеспечивает их надежной передачи, или невозможно в принципе. Поэтому на практике в качестве сигналов-носителей данных в каналах связи ВС служат или так называемые *линейные коды*, или *модулированные синусоидальные сигналы*. При этом линейные коды представляют собой двух- или многоуровневые сигналы, формируемые по определенным правилам из передаваемой последовательности, а модулированные сигналы – синусоиду, один или несколько из параметров которой (обычно частота, фаза или амплитуда в сочетании с фазой) являются некоторыми функциями от указанной последовательности. Вопросы линейного кодирования и модуляции будут подробно освещены в пп. 2.5 и 2.6.

Извлечение данных, представляемых сигналом-носителем, из этого сигнала на приемной стороне в настоящее время осуществляется методами цифровой обработки сигналов [1, 5].

Задачи *установления и разрыва соединения между абонентами*, а также *коммутации и маршрутизации данных* состоят в формировании трактов передачи данных между абонентами ВС на время сеанса связи между ними.

При организации ВС с количеством абонентов, превышающим несколько десятков, практически невозможно реализовать непосред-

ственную связь каждого из них с каждым из остальных абонентов ВС. Поэтому реальные ВС, как правило, снабжаются системой коммутационных узлов, посредством которых в процессе работы ВС могут быть сформированы тракты обмена данными между любыми двумя ее абонентами. Процедуры их формирования и носят название коммутации и маршрутизации.

Известны два основных метода формирования указанных трактов: *коммутация каналов* и *коммутация пакетов* [2, 3, 7].

Упрощенная иллюстрация метода *коммутации каналов* представлена на рис. 1.2, *а*. Сущность этого метода состоит в формировании сквозного канала обмена данными между парой абонентов через последовательность коммутационных узлов. Данный канал предоставляется в распоряжение исключительно соответствующей пары абонентов на все время сеанса связи между ними.

Сеанс связи начинается с установления соединения между абонентами. Оно осуществляется путем передачи запроса на соединение по цепи коммутационных узлов, составляющих формируемый канал связи, от абонента-инициатора сеанса обмена данными к абоненту, отвечающему на вызов (рис. 1.2, *а*). При этом, как видно из данного рисунка, длительность интервала времени между поступлением запроса в какой-либо коммутационный узел и его подачей на следующий узел, в общем случае, различна для разных узлов, из-за различного времени обработки запросов в них. По получении отвечающим абонентом запроса на соединение он отправляет абоненту-инициатору подтверждение соединения, указывающее на то, что канал связи между ними сформирован, и может быть начат собственно обмен данными, до окончания которого указанный канал поддерживается в активном (коммутированном) состоянии. Данные по каналу передаются сплошным потоком, с постоянной скоростью, определяемой звеном с минимальной пропускной способностью, практически без задержек в промежуточных узлах.

По окончании сеанса обмена данными осуществляется размыкание соединений между коммутационными узлами, образовывавшими канал, после чего эти узлы могут использоваться для формирования канала связи между какой-либо другой парой абонентов.

В свою очередь, сущность метода *коммутации пакетов* иллюстрируется рис. 1.2, *б* и упрощенно может быть описана следующим образом. Подлежащее передаче сообщение разбивается на относительно небольшие единицы информации, называемые *пакетами*, объ-

ем которых лежит в пределах, определяемых конкретным протоколом связи. Обычно нижний из них равен нескольким десяткам байт, а верхний – нескольким килобайтам [3]. Каждый из них нумеруется в соответствии с его порядковым номером в сообщении, а также снабжается информацией, необходимой для доставки пакета по назначению
(например, во многих, но не во всех практических случаях – адресом абонента-получателя). Общий принцип передачи пакетов по сети состоит в следующем.

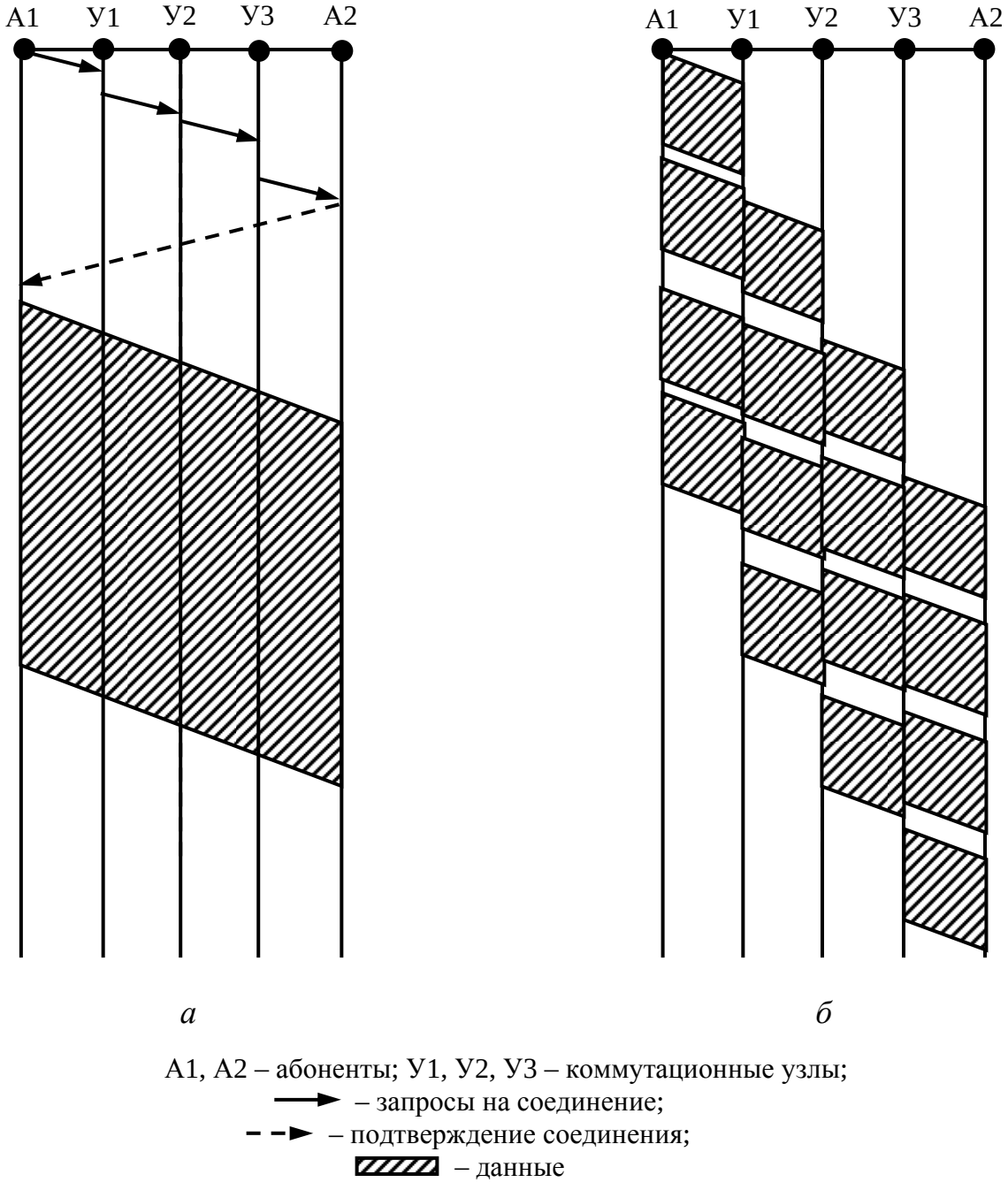


Рис. 1.2. Упрощенные иллюстрации методов коммутации каналов (*а*) и пакетов (*б*)

На основании содержащейся в пакете информации, получивший его коммутационный узел передает пакет в следующий пункт выбранного *маршрута* передачи пакетов (т. е. цепи промежуточных коммутационных узлов ВС) от этого узла к абоненту-получателю. При этом под следующим пунктом подразумевается очередной коммутационный узел, принадлежащий соответствующему маршруту. В общем случае, маршруты продвижения пакетов по сети выбираются коммутационным оборудованием, исходя из содержащейся в пакетах информации о пункте назначения, а также о требованиях к качеству их доставки. Например, для передачи пакетов компьютерного трафика, наиболее важным показателем качества доставки которых являются потери информации при продвижении пакета по сети, выбираются маршруты с минимальным уровнем ошибок обмена данными в каналах связи. В свою очередь, для продвижения по сети пакетов мультимедийного трафика, основным критерием качества передачи которых являются задержки доставки, выбираются маршруты с минимальными значениями этих задержек. Алгоритмы выбора маршрутов продвижения пакетов по сети (*маршрутизации*) будут вкратце рассмотрены ниже.

Как видно из рис. 1.2, *б*, времена обработки как различных пакетов в одном и том же коммутационном узле, так и одного и того же пакета в различных узлах, в общем случае, различны.

Процесс продвижения пакета по сети продолжается до его доставки абоненту-получателю. Объединение пакетов в исходное сообщение осуществляется на приемной стороне.

Основными отличиями метода коммутации пакетов от коммутации каналов являются следующие:

- не устанавливается сквозной канал связи между абонентами с постоянной скоростью передачи данных; физический канал связи формируется только между двумя соседними коммутационными узлами;

- данные передаются не сплошным потоком, а в виде последовательности отдельных, достаточно автономных единиц информации (пакетов), причем, в общем случае, пакеты одного и того же сообщения могут передаваться по разным маршрутам и прибывать в пункт назначения в порядке, не совпадающем с их номерами в сообщении;

- одна и та же цепь коммутационных узлов может использоваться несколькими парами абонентов для обмена данными;

- при коммутации пакетов промежуточные узлы снабжаются внутренней буферной памятью для временного хранения пакетов, так как, во-первых, решение о маршруте и параметрах дальнейшего продвижения пакета по сети принимается только по получении пакета целиком, а во-вторых – не исключено возникновение *очередей* пакетов, перемещаемых по одному и тому же маршруту.

Основными *способами* передачи пакетов по сети являются следующие [3]:

- *дейтаграммный* способ;
- формирование *логических соединений*;
- формирование *виртуальных каналов*.

Дейтаграммный способ продвижения пакетов по сети состоит в следующем. Все пакеты передаваемого потока данных продвигаются по сети независимо друг от друга, как самостоятельные единицы информации, называемые *дейтаграммами*. При этом продвижение по сети пакетов, принадлежащих к одному и тому же сообщению, в большинстве практических случаев осуществляется по различным *маршрутам* (т. е. через различные цепи промежуточных коммутационных узлов), однако, как правило, по одному и тому же *алгоритму маршрутизации*, т. е. выбора маршрута (см. далее). Важной отличительной особенностью дейтаграммного способа является то, что при обработке очередного пакета, в том числе при выборе маршрута его передачи, результаты передачи предыдущих пакетов не учитываются.

Способ передачи пакетов по сети *с формированием логических соединений*, в отличие от дейтаграммного способа, характеризуется обработкой не каждого пакета индивидуально, а всей совокупности пакетов, передаваемой между какой-либо конкретной парой абонентов, т. е. относящихся к одному и тому же *соединению*. Основным отличием совокупной обработки пакетов соединения от индивидуальной является учет «предыстории» процесса обмена данными, т. е. результатов передачи предыдущих пакетов (например, числа потерянных пакетов и т. п.) при выборе маршрута и параметров (например, скорости) передачи последующих пакетов соединения. Все пакеты соединения обслуживаются по одним и тем же правилам; однако различные пакеты одного и того же соединения, в принципе, могут передаваться по различным маршрутам. Для обработки по различным правилам па-

кетов, относящихся к различным соединениям, каждому из них присваивается индивидуальный *идентификатор*, кодом которого снабжаются все пакеты соответствующего соединения.

Следует отметить, что формирование логического соединения может осуществляться как с его *предварительным установлением*, так и без него [3].

В первом случае сеанс обмена данными начинается с передачи пакета «Запрос на установление соединения» вызывающим абонентом отвечающему. По получении последним этого пакета и при возможности установления соединения отвечающий абонент отправляет вызывающему пакет «Подтверждение установления соединения». В нем обычно указываются предлагаемые отвечающим абонентом параметры соединения (например, максимальный размер пакета и т. п.). После подтверждения соединения начинается обмен пакетами данных. По окончании сеанса обмена данными один из абонентов (обычно вызывающий) инициирует разрыв логического соединения передачей пакета «Запрос на разрыв соединения».

Формирование логического соединения без его предварительного установления не предполагает предварительного обмена пакетами запроса и подтверждения соединения между абонентами. Сеанс связи включает в себя передачу только непосредственно пакетов данных, причем обработка в коммутационных узлах каждого последующего пакета осуществляется на основании результатов передачи предыдущих пакетов (см. выше).

Нетрудно увидеть, что вариант с предварительным установлением соединения более надежен, но требует дополнительных затрат времени на обмен служебными пакетами между абонентами.

Способ *с формированием виртуальных каналов* состоит в передаче пакетов по сети от абонента-источника к абоненту-приемнику по единственному, заранее «проложенному» маршруту. Обычно «прокладка» осуществляется первым пакетом сообщения, называемым пакетом установления соединения (или пакетом-«разведчиком»). Он снабжается адресом получателя, а также специальным *идентификатором* потока данных, для которого прокладывается маршрут. Таким же идентификатором снабжаются и все остальные пакеты соответствующего потока данных. В простейшем случае, идентификатор представляет собой номер виртуального канала [3].

По мере прохождения пакета-«разведчика» по коммутационным узлам сети, на основании содержащейся в данном пакете информа-

ции, этими узлами осуществляется выбор дальнейшего маршрута его передачи, т. е. следующего коммутационного узла, в который следует направить пакет-«разведчик». Данный процесс продолжается до достижения указанным пакетом абонента-приемника. В каждом из коммутационных узлов, по которым проходит маршрут пакета установления соединения, формируется регистрационная запись соответствующего соединения (потока данных). Она содержит его идентификатор и адрес следующего коммутационного узла, в который должны направляться пакеты этого потока, т. е. пакеты, снабженные его идентификатором. Следует также отметить, что при передаче пакетов по сети методом формирования виртуальных каналов адресами отправителя и получателя обычно снабжается только пакет установления соединения, а в пакетах данных в качестве адресной информации указывается только идентификатор соединения.

Основные достоинства и недостатки трех вышеописанных способов передачи пакетов по сети представлены в табл. 1.1 [3].

Таблица 1.1

Основные достоинства и недостатки способов передачи пакетов по сети

Способ передачи пакетов	Основные достоинства	Основные недостатки
Дейтаграммный	Отсутствие каких-либо предварительных процедур перед сеансом обмена данными. Возможность независимой работы передатчика и приемника. Возможность повышения скорости обмена данными за счет передачи пакетов одного и того же сообщения по различным маршрутам	Сложность проверки передающим абонентом факта получения пакета адресатом. Вероятность потерь пакетов. Вероятность непроизводительной загрузки сети передаваемыми «вхолостую» пакетами при неготовности адресата
С формированием логических соединений	Потенциально наивысшая эффективность использования ресурсов сети. При обмене с предварительным установлением соединения – повышенная надежность за счет разрыва соединения только при полу-	Сложность реализации. Сложность разрешения ситуаций, при которых адресат не готов к приему пакетов

	чении адресатом последнего пакета сообщения	
С формированием виртуальных каналов	Наивысшая надежность обмена данными. Простота реализации	Низкая скорость обмена данными. Наименьшая эффективность использования сетевых ресурсов

Как видно из табл. 1.1, вышерассмотренные способы передачи пакетов по сети взаимно дополняют друг друга по совокупности достоинств и недостатков. Поэтому в современных сетевых технологиях и протоколах, в принципе, используются все три вышеописанных способа [3]. Ряд сетевых технологий включает в себя протоколы, основанные на двух из этих способов или на всех трех. Например, стек протоколов TCP/IP [2, 3, 7] использует протоколы передачи пакетов по сети, основанные как на дейтаграммном способе, так и на установлении логических соединений.

Процедура выбора маршрутов передачи пакетов известна под названием *маршрутизации*. Известен ряд алгоритмов маршрутизации, относящихся к одной из следующих основных групп [1]:

- статической маршрутизации;
- адаптивной маршрутизации;
- маршрутизации от источника;
- лавинной маршрутизации;
- маршрутизации, управляемой событиями.

Алгоритмы первых двух из вышеперечисленных групп базируются на применении специальных *таблиц маршрутизации*, которые создаются и ведутся в памяти коммутационных узлов ВС. Они представляют собой совокупность записей, каждая из которых содержит адрес следующего пункта маршрута, по которому из соответствующего узла должны передаваться пакеты того или иного потока данных.

Входными данными при обращении к таблицам маршрутизации являются содержащиеся в пакетах адреса получателей или идентификаторы потока данных, а также указанные в пакетах требования к качеству их доставки (см. выше).

Таблицы маршрутизации составляются исходя из некоторых критериев. Наиболее распространены на практике следующие [3]:

- критерий кратчайшего расстояния до абонента-получателя, где под расстоянием понимается число промежуточных коммутационных узлов маршрута между пунктом назначения пакетов соответствующего потока данных и текущим узлом;

- критерий максимальной пропускной способности маршрута;
- критерий минимальной задержки доставки пакетов по маршруту;
- критерий максимальной надежности доставки пакетов по маршруту (т. е. максимальной вероятности их доставки абоненту-получателю без ошибок);
- комплексные показатели качества маршрутов, учитывающие два или более критерия из числа вышеперечисленных.

При *статической маршрутизации* ее таблицы составляются и вводятся в память коммутационных узлов сетевыми администраторами, обычно вручную, и изменяются достаточно редко (минимум – раз в несколько дней). При значительно более распространенной на практике *адаптивной маршрутизации* ее таблицы ведутся автоматически, под управлением специального программного обеспечения, в соответствии с текущим состоянием каналов связи ВС (текущих значений пропускной способности каждого из потенциальных маршрутов передачи пакетов, текущих значений задержки доставки пакетов по каждому из них и т. п.). Сбор информации о состоянии каналов связи осуществляется путем обмена специальными тестовыми пакетами между коммутационными узлами ВС в соответствии с определенными протоколами сбора маршрутной информации. Известен ряд таких протоколов. В частности, на практике широко распространены протоколы, основанные на *дистанционно-векторных алгоритмах* (*Distance Vector Algorithms, DVA*) и *алгоритмах состояния связей* (*Link State Algorithms, LSA*). Первая из названных групп алгоритмов основывается на периодическом сборе каждым из коммутационных узлов информации об *условных расстояниях* до остальных узлов ВС. Под условным расстоянием при этом понимается параметр маршрута, прямо пропорциональный числу его промежуточных коммутационных узлов и обратно пропорциональный его пропускной способности. Алгоритмы группы LSA базируются на сборе информации о состоянии линий связи по некоторым критериям (задержкам доставки пакетов, среднему количеству ошибок на пакет при его передаче по тому или иному маршруту и т. п.).

Алгоритмы *маршрутизации от источника*, *лавинной маршрутизации* и *маршрутизации*, *управляемой событиями*, в отличие от ранее рассмотренных, не предполагают использования таблиц маршрутизации.

Маршрутизация от источника (*Source Routing*, *SR*) состоит в указании в каждом из пакетов заранее определенного маршрута его продвижения по сети, т. е. последовательности промежуточных коммутационных узлов, через которые он должен быть передан абоненту-получателю. Маршрут или задается вручную сетевым администратором, или определяется автоматически узлом – отправителем пакета на основании данных о конфигурации и состоянии каналов связи ВС. Маршрутизация от источника ограниченно использовалась на ранних этапах развития ГВС (в частности, Интернет) [3, 7]. В настоящее время она применяется крайне редко, так как при ней весьма сложно обеспечить эффективное использование каналов связи ВС.

Принцип *лавинной маршрутизации* заключается в том, что по получении пакета некоторым коммутационным узлом он передает этот пакет всем соседним с ним по сети узлам, кроме того, из которого данный пакет был получен. Этот принцип прост в реализации, однако неэффективен с точки зрения использования каналов связи ВС. В самом деле, при лавинной маршрутизации один и тот же пакет параллельно передается по множеству каналов. Тем не менее, в настоящее время лавинная маршрутизация все же ограниченно применяется на практике. В частности, она используется в ЛВС при продвижении по сети блоков данных с неизвестным адресом получателя.

Маршрутизация, управляемая событиями (*Event Dependent Routing*, *EDR*) состоит в передаче пакета по маршруту, уже приводившему ранее к успеху при пересылке пакетов того же потока данных (т. е. с тем же адресом назначения). Например, согласно одному из типовых алгоритмов маршрутизации, относящихся к данной группе [3], перед отправлением пакета данных из некоторого коммутационного узла им во все соседние узлы соответствующего направления передаются запросы соединения, а пакет данных направляется в узел, подтверждение соединения от которого пришло первым. Алгоритмы маршрутизации, относящиеся к данной группе, применялись на ранних этапах развития Интернет [3, 7]; в настоящее время они мало распространены на практике.

В целом, наиболее широко применяемыми в настоящее время являются алгоритмы адаптивной маршрутизации [3].

Подробное описание алгоритмов и протоколов маршрутизации будет представлено в гл. 4.

Следует также отметить, что, кроме методов коммутации каналов и коммутации пакетов, теоретически возможно осуществлять обмен данными между отправителем и получателем также методом *коммутации сообщений*. Он аналогичен методу коммутации пакетов, за исключением того, что подлежащее передаче сообщение продвигается по сети целиком, без его разбиения на пакеты. Коммутация сообщений не нашла широкого распространения на практике, в первую очередь – из-за сложности их эффективной маршрутизации и буферизации, обусловленной широким диапазоном, в котором может находиться объем сообщений даже одного и того же потока данных. Поэтому в дальнейшем метод коммутации сообщений рассматриваться не будет.

Для дальнейшего изложения материала необходимо сопоставить между собой методы коммутации каналов и пакетов с целью выявления достоинств и недостатков каждого из них. Основные свойства указанных методов в соответствии с их вышеизложенными описаниями представлены в табл. 1.2.

Таблица 1.2

Основные свойства методов коммутации каналов и пакетов

Свойство	Метод коммутации	
	Коммутация каналов	Коммутация пакетов
Необходимость формирования физического канала связи между абонентами	Есть	Нет
Возможность организации нескольких параллельно действующих маршрутов передачи данных между абонентами	Нет	Есть
Возможность одновременного использования одного и того же участка некоторого маршрута несколькими парами абонентов	Нет	Есть
Готовность сети к приему данных	Только по установлении соединения	Постоянно
Критичность выхода из строя одного из коммутационных узлов	Да	Нет
Необходимость промежуточного хранения данных в коммутационных узлах	Нет	Есть
Эффективность использования ресурсов сети при передаче трафика реального времени*	Высокая	Низкая
Эффективность использования ресурсов сети	Низкая	Высокая

при передаче «эластичного» трафика**		

* Под трафиком реального времени понимаются потоки данных, которые должны передаваться с минимально возможными задержками, например, голосовой трафик в Интернет-телефонии или видео- и аудиоданные в Интернет-телевидении.

** Под «эластичным» трафиком подразумеваются потоки данных, допускающие задержки их доставки, но не допускающие потерь информации. К потокам данных этого типа относятся практически все разновидности компьютерного трафика (см. подп. 1.1.3), например, передача текстовых и/или графических документов научно-технического содержания, электронная почта и т. п.

На основании содержащихся в табл. 1.2 данных, можно сделать следующие выводы. Метод коммутации каналов, в общем, предпочтителен только при передаче трафика реального времени. Для компьютерного трафика, а также для тех разновидностей мультимедийного трафика, которые некритичны к задержкам доставки информации (например, видеоданных, скачиваемых для дальнейшего просмотра
в режиме off-line), более предпочтителен метод коммутации пакетов. Поэтому в целом он более распространен в практике современных сетевых технологий. При этом на практике используются все три рассмотренных способа передачи пакетов по сети: дейтаграммный, с формированием логических соединений и с формированием виртуальных каналов. Из алгоритмов маршрутизации пакетов наибольшее распространение получили алгоритмы адаптивной маршрутизации.

В дальнейшем, при изложении материалов данного учебного пособия будет предполагаться, что формирование трактов обмена данными между абонентами ВС осуществляется методом *коммутации пакетов* (если не оговаривается иное).

1.2.2. Выбор рациональной структуры ВС и конфигурации связей между ее абонентами. Адресация абонентов

Как было указано ранее, структура и топология ВС, а также система адресации ее абонентов должны обеспечивать рациональное сочетание производительности, надежности, расширяемости и масштабируемости ВС.

Простейшим вариантом структуры ВС является *моноканал* [3]. Он представляет собой совокупность абонентов, каждому из которых присвоен индивидуальный адрес, соединенных между собой линиями связи таким образом, чтобы обеспечить возможность обмена данными между любыми двумя абонентами ВС. Наиболее распространенными

вариантами топологии моноканала являются полносвязная, шинная, кольцевая и звездообразная, представленные на рис. 1.3 [3].

Полносвязная топология (рис. 1.3, *а*) предполагает связь каждого из абонентов с каждым по отдельной линии. В моноканале с шинной топологией (рис. 1.3, *б*) все его абоненты параллельно подключены к одной и той же линии связи, совместно используемой ими для обмена данными. При кольцевой топологии моноканала (рис. 1.3, *в*) каждый из его абонентов соединен с двумя соседними, а связь между какими-либо двумя абонентами осуществляется через цепочку включенных между ними последовательно соединенных абонентов. Звездообразная топология (рис. 1.3, *г*) предполагает наличие в моноканале центрального распределительного узла (на рис. 1.3, *г* – изображен в центре), через который осуществляется обмен данными между любыми двумя абонентами. Функции таких узлов в настоящее время обычно выполняют коммутаторы, принципы построения и применения которых будут изложены далее, в п. 3.7.

Следует отметить, что представленные на рис. 1.3 топологии показаны весьма упрощенно. На практике для формирования моноканала определенного типа (шинного, кольцевого и т. п.) применяются специальные устройства, вопросы реализации и использования которых также будут изложены далее, в п. 3.7.

Из вышеописанных топологий *полносвязная* обеспечивает наивысшую производительность обмена данными между абонентами, однако, с другой стороны, она наиболее сложна в реализации. Поэтому она редко применяется на практике. *Звездообразная* топология характеризуется относительно низкой надежностью, обусловленной тем, что при выходе из строя центрального распределительного узла обмен данными между абонентами становится невозможным. Однако в настоящее время существуют эффективные средства минимизации вероятности возникновения такой ситуации [3]. *Кольцевая* топология обладает более высокой надежностью, чем звездообразная, так как при выходе из строя какого-либо из абонентов и, следовательно, невозможности передачи данных через него, они могут пересылаться по кольцу в противоположном направлении, например, не «по часовой стрелке», а «против» (см. рис. 1.3, *в*).

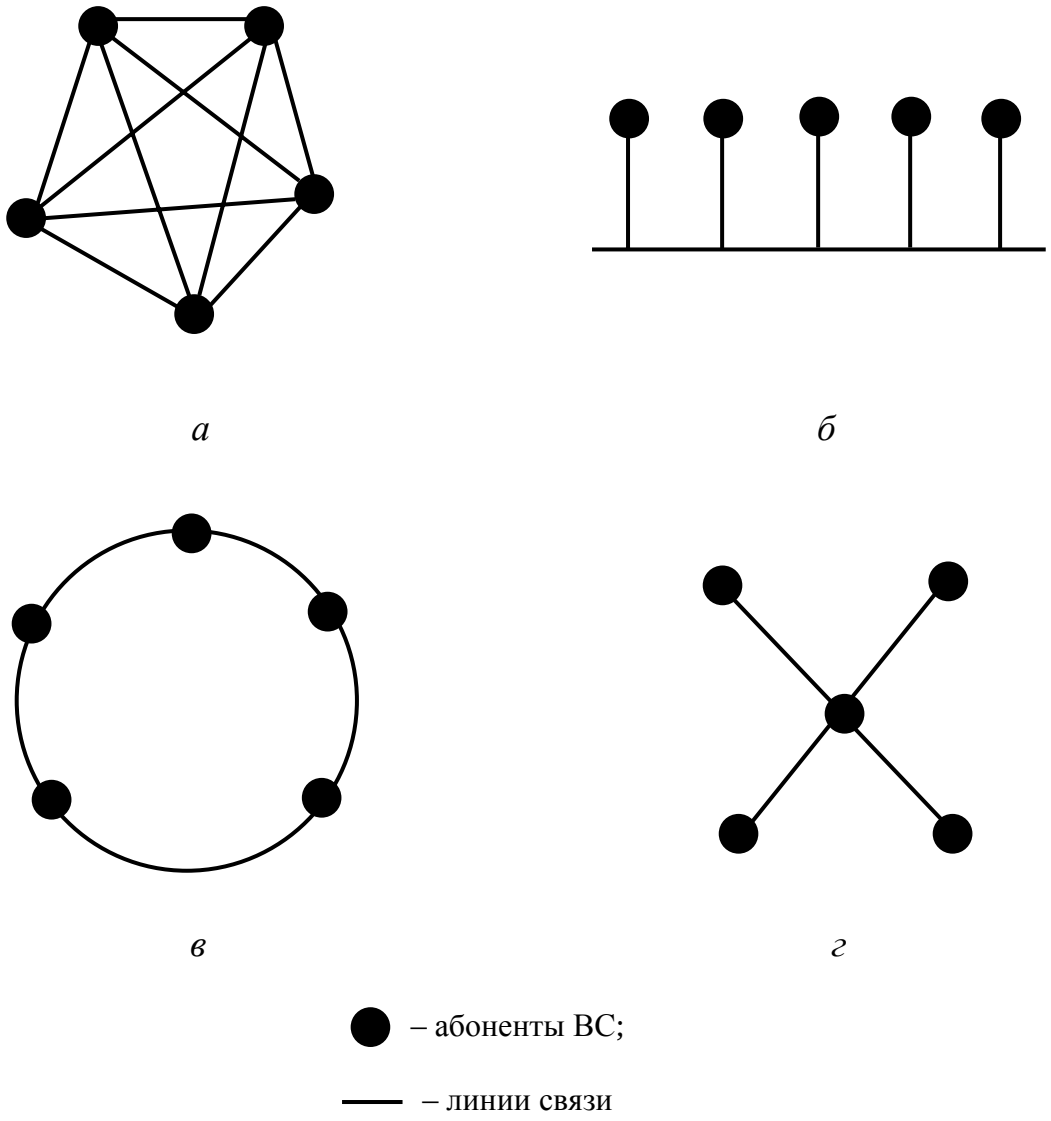


Рис. 1.3. Распространенные варианты топологии моноканала ВС:
а – полносвязная; *б* – шинная; *в* – кольцевая; *г* – звездообразная

Однако кольцевая топология характеризуется меньшей производительностью, чем шинная (см. далее), за счет необходимости передачи данных между абонентами, в общем случае, через последовательность промежуточных узлов. Поэтому «кольцо», будучи достаточно широко распространенной разновидностью топологии моноканала на ранних этапах развития ВС, в настоящее время находит относительно ограниченное применение [2, 3]. Основным недостатком *шинной* топологии являются конфликты абонентов из-за доступа к совместно используемой ими линии связи. Однако существуют достаточно эффективные способы разрешения данных конфликтов, описанные далее, в п. 3.4. В целом, в настоящее время наиболее распространенной топологией монокана-

ла является *звездообразная с физической точки зрения* (с коммутатором в качестве центрального узла), но при этом *шинная с логической точки зрения*, т. е. с точки зрения взаимодействия абонентов [3]. Такой подход характерен, например, для современных технологий Ethernet [3].

Более детально основы реализации моноканалов ВС будут рассмотрены в гл. 3.

Однако количество абонентов моноканала достаточно ограничено. В самом деле, при числе абонентов шинного моноканала, большем некоторого порогового значения (на практике – нескольких десятков), производительность обмена данными между ними резко снижается из-за частых конфликтов и очередей при доступе к совместно используемой линии связи. То же имеет место при кольцевой топологии моноканала. Звездообразная топология моноканала также характеризуется резким снижением производительности при числе абонентов, большем некоторой пороговой величины, обусловленным очередями данных в центральном распределительном узле.

Ввиду вышесказанного, по структуре моноканала реализуются только наиболее простые ВС (ЛВС с количеством абонентов до нескольких десятков). При числе абонентов ВС, большем нескольких десятков, ее реализация в виде моноканала является нерациональной. Одним из наиболее распространенных подходов к построению таких ВС является их реализация в виде *сегментированных коммутируемых ВС*. Они представляют собой совокупность связанных между собой *сегментов*, т. е. моноканалов с определенной топологией, например, шинной, и с числом абонентов, не более некоторого задаваемого конкретными стандартами ВС. При этом каждому из абонентов присваивается индивидуальный адрес в пределах всей сети (а не отдельного сегмента), т. е. сегментированная ВС, как и моноканал, характеризуется *одноуровневой* системой адресации. Все сегменты используют одни и те же стандарты (протоколы) кодирования, оформления и представления данных. Связь между сегментами осуществляются посредством специальных устройств, наиболее распространенными из которых являются *коммутаторы* (*switches*) [3]. Применяются и другие типы коммутационных устройств [2, 3, 7]. Их основной функцией является передача данных только в тот сегмент, в котором находится абонент-получатель. Это «разгружает» другие сегменты от не предназначенных им данных и, как следствие, повышает производительность ВС в целом. Коммутационные устройства функционируют на основе хранящихся в их памяти *адресных таблиц*,

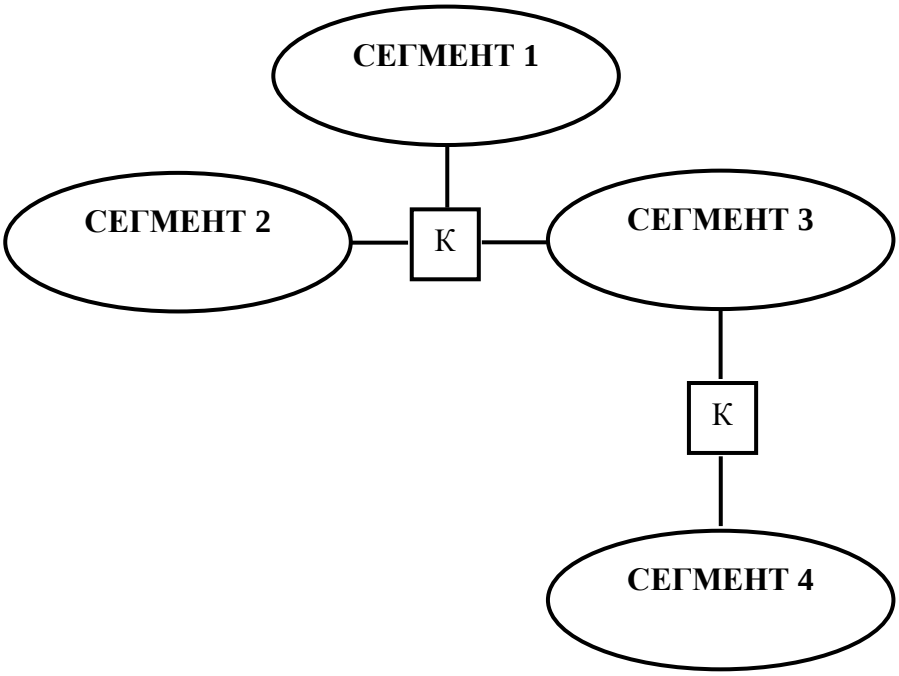
устанавливающих соответствие между адресом абонента и номером сегмента, в котором он находится. Эти таблицы создаются или сетевым администратором, или (чаще) автоматически, в процессе «самообучения» [3].

Пример упрощенной структуры фрагмента сегментированной коммутируемой ВС представлен на рис. 1.4.

Более детально принципы реализации сегментированных коммутируемых ВС будут описаны в гл. 3.

В виде сегментированных коммутируемых сетей часто строятся ЛВС. Во многих практических случаях такой подход достаточен для обеспечения их приемлемой производительности. Однако при построении ГВС, а в ряде случаев – и крупных ЛВС, только разбиение ВС на коммутируемые сегменты является недостаточным для ее эффективной реализации. Основными причинами этого являются следующие [3]:

- низкая пропускная способность и надежность сегментированных коммутируемых ВС при достаточно большом количестве сегментов, обусловленная необходимостью существования *единственного* пути между любыми двумя абонентами такой ВС для корректной работы ее коммутационных устройств, в то время как для обеспечения приемлемых пропускной способности и надежности достаточно крупной ВС необходимо существование *нескольких* таких путей [3];



К – коммутаторы

Рис. 1.4. Пример упрощенной структуры фрагмента сегментированной коммутируемой ВС

- сложность или невозможность объединения в составе сегментированной коммутируемой ВС фрагментов с различными стандартами (протоколами) кодирования, оформления и представления данных, в то время как существование таких фрагментов в пределах ГВС или крупной ЛВС практически неизбежно.

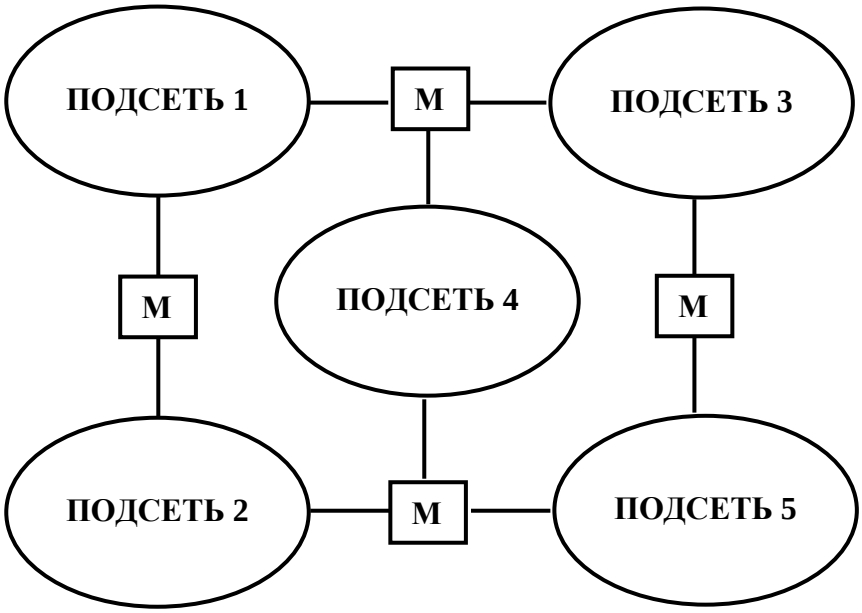
Поэтому ГВС и, в ряде практических случаев, достаточно крупные ЛВС, обычно реализуются в виде *составных сетей* (*интерсетей*), в англоязычной литературе – *internetworks* или *internets* (не путать с названием Всемирной сети *Internet*).

Упрощенная структура фрагмента интерсети представлена на рис. 1.5.

Интерсеть строится как совокупность взаимно связанных *подсетей* (*subnets*), каждая из которых, в свою очередь, обычно представляет собой сегментированную ЛВС на основе коммутаторов (см. рис. 1.4). В общем случае, подсети используют различные стандарты (протоколы) кодирования, оформления и представления данных, однако эти стандарты (протоколы) едины в пределах каждой из подсетей. Для обмена данными между ними служат устройства, называемые *маршрутизаторами* (в англоязычной литературе – *routers*).

В литературе их также часто называют *шлюзами* (*gateways*), хотя, строго говоря, данное название применимо только к маршрутизаторам, соединяющим между собой подсети, реализованные в соответствии с различными стандартами [2, 3, 7]. Маршрутизаторы выполняют следующие основные функции [1]:

- выбор маршрутов передачи данных между подсетями, наиболее рациональных по некоторым критериям (см. подп. 1.2.1);



М – маршрутизаторы

Рис. 1.5. Упрощенная структура фрагмента интерсети

- взаимное преобразование (трансляция) данных при обмене ими между подсетями с различными стандартами (протоколами) их кодирования, оформления и представления.

Как правило, маршрутизаторы строятся на основе специализированных компьютеров, снабженных интерфейсными блоками для подключения к подсетям, а также программным обеспечением, реализующим процедуры маршрутизации и преобразования данных. Реже маршрутизаторы реализуются в виде программных моделей на компьютерах общего назначения.

В составных ВС, в отличие от ранее рассмотренных, применяется *двухуровневая* система адресации. Каждому из абонентов ВС присваиваются два адреса:

- *сетевой* адрес, уникальный (не повторяющийся) в пределах сети в целом, назначаемый сетевым администратором и обычно состоящий из двух частей – номера подсети и номера абонента;

- *локальный* (*аппаратный*) адрес, действующий только в пределах подсети, в которой находится абонент; данный адрес называется аппаратным, так как обычно он присваивается сетевому оборудованию абонентов при его изготовлении.

При этом следует отметить, что каждый из маршрутизаторов является абонентом нескольких подсетей и, как следствие, каждому из его портов присваиваются собственный сетевой и локальный адреса.

Процесс обмена данными между абонентами составной сети осуществляется следующим образом. Пакеты (см. подп. 1.2.1) при их передаче по каждой из подсетей оформляются маршрутизаторами в кадры, обобщенный формат которых представлен на рис. 1.6.

Рис. 1.6. Обобщенный формат кадра подсети, входящей в составную сеть

Пакет инкапсулируется («вкладывается») в кадр в качестве его поля данных, аналогично тому, как письмо вкладывается в конверт. В свою очередь, роль «конверта» при этом играют заголовок и концевик кадра, оформляемые в соответствии со стандартами подсети, в которую он направляется. Заголовки как кадра, так и пакета включают в себя адресную информацию, однако различных уровней. Заголовок пакета содержит сетевые адреса его отправителя и получателя, т. е. начального и конечного пунктов его маршрута по сети в целом, а заголовок кадра – локальные адреса маршрутизатора-отправителя и маршрутизатора-получателя пакета в пределах подсети, в которую он направляется, т. е. начального и конечного пунктов его продвижения по этой подсети. При этом локальный адрес получателя определя-

ется маршрутизатором-отправителем в соответствии с указанными в заголовке пакета сетевыми адресами абонента-отправителя и получателя, применяемым способом передачи пакетов, а также алгоритмом маршрутизации

После оформления кадра он представляется сигналом-носителем в соответствии со стандартами подсети, в которую он направляется, и передается маршрутизатору-получателю. Последний извлекает из кадра содержащийся в нем пакет и, на основании имеющейся в заголовке пакета адресной и служебной информации, а также используемого способа маршрутизации, определяет подсеть, в которую следует направить пакет, и локальный адрес маршрутизатора - получателя пакета в этой подсети. Затем он оформляется в кадр в соответствии со стандартами указанной подсети, и направляется очередному маршрутизатору-получателю. Данный процесс продолжается до достижения пакетом конечного пункта назначения.

Детальное изложение основ реализации составных сетей будет представлено в гл. 4.

Основными преимуществами составной сети, по сравнению с более простыми вариантами реализации ВС (сегментированной сетью и моноканалом), являются следующие:

- возможность использования различных сетевых технологий (стандартов, протоколов) в рамках сети, что является необходимым условием реализации крупных ВС, в частности, ГВС;

- возможность организации нескольких параллельных маршрутов передачи данных между двумя абонентами ВС, что повышает производительность и надежность сети.

Благодаря этим преимуществам, принцип составной сети в настоящее время является наиболее распространенным при реализации достаточно крупных ЛВС, а также ГВС (в том числе Интернет).

С другой стороны, составные ВС характеризуются относительно высокой сложностью. Поэтому применение технологий составных сетей не оправдано при построении сравнительно простых ЛВС, однородных с точки зрения используемых сетевых стандартов (протоколов). Такие ЛВС, как правило, реализуются в виде сегментированных коммутируемых ВС или (реже) моноканалов.

В заключение следует отметить, что кроме вышеописанных базовых структурных решений ВС, известен и ряд других [3], а представленное выше изложение принципов структурно-топологической орга-

низации ВС является весьма упрощенным. Детальное описание указанных принципов будет приведено в гл. 3 и 4.

1.2.3. Обеспечение совместной работы в составе ВС разнотипных и разнородных средств. Организация «прозрачного» доступа абонентов ВС к общесетевым ресурсам

Решение данных проблем осуществляется путем реализации сетевых аппаратных средств и сетевого программного обеспечения (ПО) по принципу *открытых систем*, т. е. в соответствии с *открытыми*, общедоступными стандартами, протоколами и рекомендациями, едиными в пределах сети или подсети. Это позволяет без существенных материальных и временных затрат объединять в составе одной и той же ВС разнотипные (в том числе от разных производителей) сетевые аппаратные средства, коммутационное оборудование и компьютеры, а также обеспечивать корректное взаимодействие их ПО. Кроме того, при взаимной совместимости сетевых аппаратных и программных средств по стандартам / протоколам и рекомендациям, процессы установления связи между абонентами ВС, преобразования и продвижения данных по сети реализуются автоматически и «незаметно» или «малозаметно» для пользователей ВС, т. е. имеет место их «прозрачный» доступ к сетевым ресурсам.

Вообще говоря, для построения работоспособной ВС необходимо обеспечить взаимную совместимость сетевых аппаратных и программных средств по весьма большому количеству разнообразных стандартов и рекомендаций [2, 3, 7]. При этом некоторые из них должны соблюдаться в пределах всей сети, некоторые – только в пределах одной подсети, одни должны поддерживаться пользовательским ПО, другие – маршрутизаторами, и тому подобное. Таким образом, указанные стандарты / протоколы и рекомендации должны быть сгруппированы и упорядочены в зависимости от функций обмена данными, реализация которых обеспечивается соблюдением тех или иных протоколов и/или рекомендаций. Их группирование и упорядочивание осуществляется на основе иерархических многоуровневых *моделей взаимодействия* абонентов ВС в процессе обмена данными.

Указанные модели являются одним из базовых вопросов теории и практики ВС, поэтому ему посвящается отдельный пункт (см. далее).

1.3. Модели взаимодействия абонентов ВС

Модель взаимодействия абонентов ВС, в общем случае, представляет собой иерархическую, упорядоченную последовательность процедур, описывающую процесс обмена данными между указанными абонентами.

Как следует из вышесказанного, этот процесс, в общем случае, включает в себя следующие основные процедуры:

- установление связи между абонентами, ее поддержка в течение сеанса обмена данными и разрыв связи по его окончании;
- получение одним абонентом (с помощью пользовательского программного обеспечения) доступа к сетевым ресурсам другого абонента, например, к представленной в электронном виде технической документации на продукцию некоторой фирмы, хранящейся на сервере этой фирмы;
- представление абонентом-отправителем запрашиваемых данных в форме, пригодной для передачи по ВС (в том числе их кодирование, сжатие и, при необходимости, шифрование для защиты от несанкционированного доступа);
- разбиение потока данных, представляющих запрашиваемый электронный документ, на пакеты (см. подп. 1.2.1);
- определение маршрута (маршрутов) передачи пакетов абоненту-получателю;
- продвижение каждого из пакетов по маршруту через последовательность подсетей, в том числе оформление пакета в кадр и его представление сигналом-носителем в соответствии со стандартами конкретной подсети, передача данного сигнала по подсети в следующий пункт маршрута, восстановление кадра из сигнала-носителя, извлечение пакета из кадра и направление его в следующую подсеть маршрута;
- объединение пакетов на приемной стороне в единый поток данных;
- преобразование принятых абонентом-получателем данных в форму, «понятную» его пользовательским программам (декодирование, распаковка, при необходимости – дешифрование).

Следует отметить, что данная последовательность процедур, в общем случае, характерна не только для ВС, но и для других распределенных систем обмена данными.

Вообще говоря, известны несколько разновидностей моделей взаимодействия абонентов таких систем [1, 2, 3, 7], описывающих вышеперечисленную последовательность процедур. Наиболее популярной из них является предложенная Международной организацией по стандартизации (*International Standards Organization, ISO*) в 1978 г. *базовая эталонная модель взаимодействия открытых систем* (*Open Systems Interconnection Basic Reference Model, OSI*), обычно называемая *моделью OSI*. Модель OSI включает в себя семь уровней реализации процесса взаимодействия абонентов коммуникационной системы [2, 3, 7]:

- физический (*physical layer*);
- канальный (*data link layer*);
- сетевой (*network layer*);
- транспортный (*transport layer*);
- сеансовый (*session layer*);
- представительский (*presentation layer*);
- прикладной (*application layer*).

При этом модель OSI предполагает, что формирование трактов обмена данными осуществляется методом *коммутации пакетов*. Многоуровневые модели взаимодействия открытых систем с коммутацией каналов также существуют [1, 2, 3], однако их рассмотрение выходит за рамки настоящего учебного пособия. Интересующиеся лица могут ознакомиться с ними, например, по источнику [3].

Функции уровней модели OSI при обмене данными между абонентами ВС с коммутацией пакетов иллюстрирует рис. 1.7. Для определенности положено, что абонент 2 является пользовательским ПК, запрашивающим информацию от абонента 1, роль которого играет сервер некоторой фирмы. При этом, во избежание загромождения рис. 1.7, на нем показан только поток данных, получаемых от абонента 1 абонентом 2 в ответ на его запрос.

С пользовательской точки зрения, для обмена данными между абонентами, в первую очередь, необходимы программные средства, обеспечивающие:

- доступ абонента 2 к требующейся ему информации, размещенной на сервере абонента 1 (например, поиск и открытие Internet-сайта абонента 1 и запрашиваемой Web-страницы на этом сайте);

- активизацию абонентом 1 процесса передачи абоненту 2 запрашиваемой им страницы (средствами уровней с представительского по физический включительно);

- контроль процесса передачи на пользовательском уровне, в том числе отображение прогресса загрузки данных, а также формирование соответствующих сообщений («Выполнено», «Выполнено, но с ошибками на странице», «Невозможно отобразить страницу» и т. п.).

Вышеперечисленные функции относятся к *прикладному* уровню модели OSI. Они реализуются сетевым программным обеспечением (в основном – пользовательским). При этом очевидно, что средства реализации функций прикладного уровня абонентов 1 и 2 должны быть *совместимы* друг с другом для обеспечения их корректного взаимодействия. Это означает, что они должны поддерживать одни и те же *протоколы/стандарты* прикладного уровня модели OSI. К таковым относится, например, HTTP (Hyper Text Transfer Protocol), регламентирующий передачу по сети *гипертекстовых* документов, т. е. Web-страниц, представленных HTML-кодом [3]. Естественно, известен и ряд других протоколов прикладного уровня [3, 7].

Однако наличие у абонентов взаимно совместимых программных средств, реализующих перечисленные ранее функции, как несложно увидеть, не является достаточным для удовлетворения запроса абонента 2. Для этого необходимо выполнение еще нескольких условий.

Наиболее очевидным из них является представление запрашиваемых данных (например, содержимого Web-страницы, передаваемой абонентом 1 в ответ на запрос абонента 2) при передаче по сети в форме, обеспечивающей:

- корректное декодирование и представление данных на приемной стороне, в том числе при использовании различных стандартов / протоколов представления данных на сервере абонента 1 и на ПК абонента 2;

- минимальный объем при передаче по системе связи ВС (например, компрессию представленных на странице изображений);

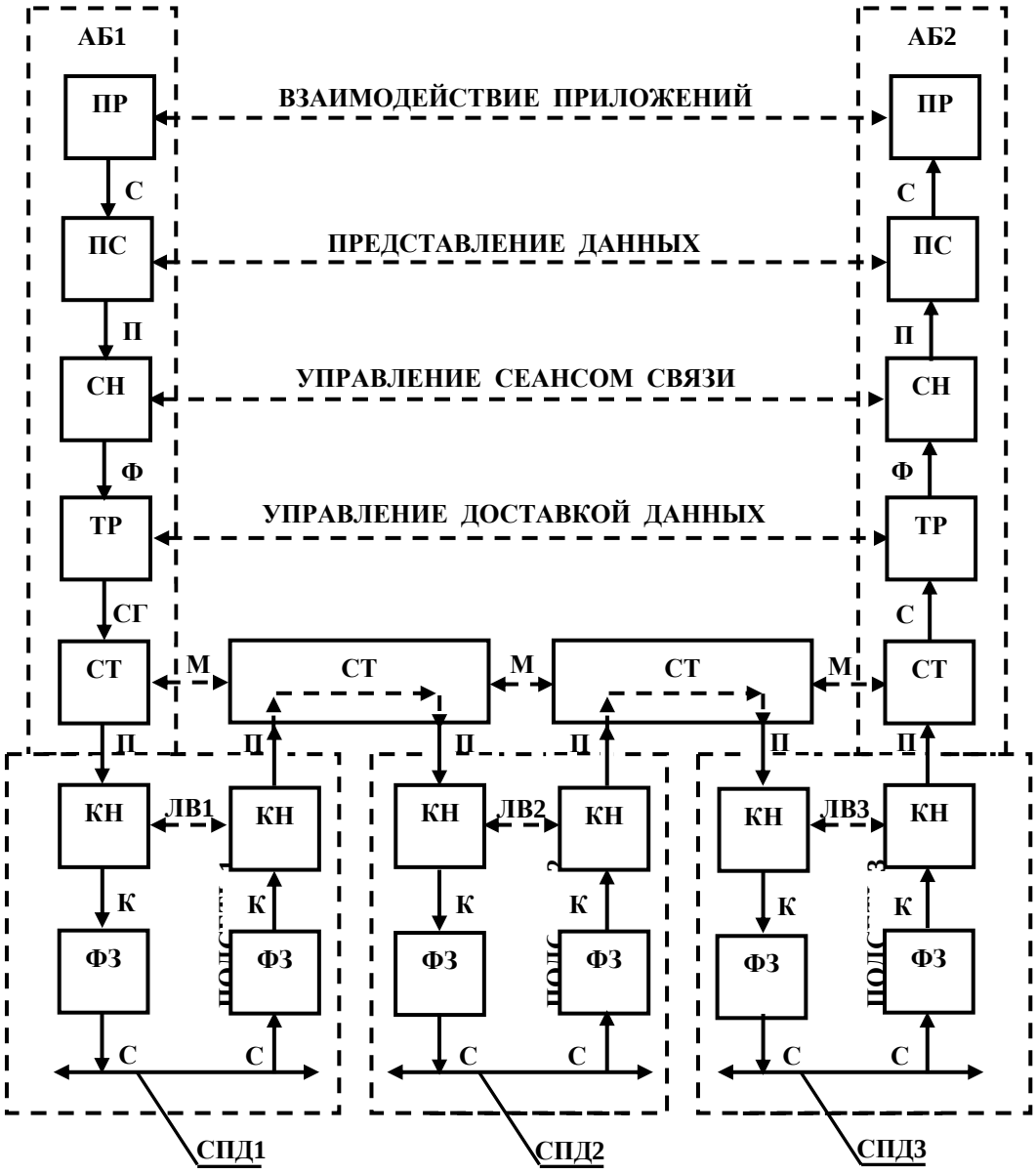
- при необходимости – защиту от несанкционированного доступа при передаче.

Вышеперечисленные условия обеспечиваются стандартами / протоколами *представительского* уровня модели OSI. В частности, к ним относятся [2, 3, 7]:

- NetWare Core Protocol (NCP), Network Data Representation (NDR), External Data Representation (XDR) и ряд других стандартов / протоколов, регламентирующих представление данных при передаче по сети;

- стандарты группы JPEG, MPEG и др., оговаривающие способы и алгоритмы сжатия (компрессии) данных;

- протокол SSL (Secure Socket Layer), обеспечивающий секретный обмен сообщениями для протоколов прикладного уровня.



АБ1, АБ2 – абоненты ВС;
ФЗ – физический уровень;
КН – канальный уровень;
СТ – сетевой уровень;
ТР – транспортный уровень;
СН – сеансовый уровень;
ПС – представитель уровень;
ПР – прикладной уровень;

СПДi – среда передачи данных i-ой подсети;
Сi – сигнал-носитель данных i-ой подсети;
Кi – кадры i-ой подсети;
ЛВi – логическое взаимодействие абонентов i-й подсети;
П – пакеты;
М – маршрутизация;
СГ – сегменты;
ФР – фрагменты;
ПД – потоки данных;
СБ – сообщения.

Рис. 1.7. Модель OSI (применительно к ВС)

При этом, чтобы данные, передаваемые прикладным уровнем абонента 1, были «понятны» прикладному уровню абонента 2, их средства реализации функций представительского уровня, очевидно, должно поддерживать одни и те же стандарты / протоколы.

Но представление запрашиваемых данных в форме, удовлетворяющей ранее перечисленным требованиям, также не является достаточным условием корректного обслуживания запроса абонента 2. Очевидно, запрашиваемые данные должны быть *доставлены* получателю. Доставка обеспечивается средствами реализации уровней модели OSI с сеансового по физический включительно.

Протоколы/стандарты *сеансового* уровня и средства их реализации обеспечивают:

- разделение потока данных, поступающего с представительского уровня, на фрагменты, передаваемые в течение одного сеанса связи;
- установление соединения между абонентами;
- определение, какой из них является инициатором соединения (вызывающим абонентом), а какой – отвечающим на вызов;
- поддержку соединения в течение сеанса связи (в том числе в течение интервалов времени, когда пользовательские сетевые приложения не активны);
- синхронизацию процесса обмена данными (путем помещения в поток данных контрольных точек, начиная с которых возобновляется процесс обмена при его нарушении);
- разрыв соединения по окончании данного сеанса.

Естественно, для корректного установления и поддержки связи между абонентами необходимо, чтобы их средства реализации сеансового уровня модели OSI поддерживали одни и те же протоколы. Примерами таких протоколов являются [2, 3, 7]: ISO-SP (OSI Session Layer Protocol), NetBIOS (Network Basic Input Output System), PPTP (Point-to-Point Tunneling Protocol), RPC (Remote Procedure Call Protocol), RTCP (Real-time Transport Control Protocol).

Следует при этом отметить, что на практике сеансовый уровень достаточно редко реализуется специально созданными для этого протоколами / средствами. Чаще, функции этого уровня объединяются с функциями прикладного уровня и реализуются в рамках одних и тех же протоколов [3].

В целом, протоколы / стандарты прикладного, представительского и сеансового уровней и средства их реализации относятся к *подсистеме предоставления прикладных сервисов* ВС.

По установлении соединения начинается собственно процесс передачи запрашиваемых данных, реализуемый на основе протоколов / стандартов уровней с транспортного по физический включительно, которые относятся к *транспортной подсистеме* ВС.

Протоколы / стандарты *транспортного* уровня и средства их реализации обеспечивают управление доставкой запрашиваемых данных от отправителя получателю. При этом, образно говоря, транспортный уровень «не интересуется» маршрутами продвижения данных по сети, они определяются на сетевом уровне (см. далее). Транспортный же уровень «интересует» только «конечный результат»: доставка данных получателю без ошибок, потерь и дублирования, в той последовательности, в которой они были переданы. Для достижения этого результата на транспортном уровне решаются следующие основные задачи:

- разбиение передаваемого сообщения на сегменты (каждый из которых оформляется в пакет на сетевом уровне);
- снабжение сегментов информацией о требованиях к качеству их доставки (доставка с минимальной задержкой; доставка с максимальной надежностью, в том числе с подтверждением получения и т. п.), а также информацией, обеспечивающей корректное объединение сегментов в исходный поток данных (в простейшем случае – их нумерация);
- передача сегментов сетевому уровню для их оформления в пакеты и дальнейшей передачи по сети;
- прием сегментов с сетевого уровня на стороне получателя, контроль правильности их доставки и объединение сегментов в исходный поток данных.

К наиболее известным протоколам транспортного уровня относятся [2, 3, 7]: TCP (Transmission Control Protocol), предполагающий передачу пакетов по сети с установлением логического соединения между абонентами, и UDP (User Datagram Protocol), предусматривающий дейтаграммный способ продвижения пакетов. Естественно, известны и другие протоколы транспортного уровня [3]. При этом, как и на ранее рассмотренных уровнях, для обеспечения корректного обмена данными между абонентами ВС их средства реализации транспортного уровня модели OSI должны поддерживать одни и те же протоколы.

Следует отметить, что уровни модели OSI, с прикладного по транспортный включительно, являются *сквозными*, т. е. на этих уровнях связь осуществляется непосредственно между сетевым ПО або-

нентов (см. рис. 1.7). Функции данных уровней также реализуются практически исключительно указанным ПО.

В свою очередь, на сетевом, канальном и физическом уровнях реализуются функции продвижения данных через последовательность коммутационных узлов коммуникационной системы ВС. Поэтому говорят, что протоколы вышеперечисленных уровней и реализующие их средства действуют на основе принципа звеньев составной цепи. При этом, в отличие от сквозных уровней, для реализации функций сетевого, канального и физического уровней необходимо не только соответствующее ПО, но и специальное сетевое оборудование: маршрутизаторы, коммутаторы, модемы, сетевые адаптеры и т. п. (см. далее).

Основной функцией средств реализации *сетевого* уровня модели OSI является обеспечение передачи данных через составную сеть (см. рис. 1.5 и пояснения к нему), в том числе [2, 3, 7]:

- оформление каждого из сегментов, поступающих с транспортного уровня, в пакет, содержащий соответствующий сегмент в качестве поля данных, а также сетевые адреса отправителя и получателя, требования к доставке пакета (наименьшая задержка, наивысшая надежность и т. п.) и другую служебную информацию в качестве заголовка (см. рис. 1.6, а также гл. 4);

- сбор информации о текущем состоянии маршрутов (пропускной способности, уровне ошибок передачи и т. п.) с целью выбора наиболее рациональных маршрутов с точки зрения требований к качеству доставки пакетов;

- определение маршрутов продвижения пакетов по сети на основании применяемого способа их передачи и алгоритмов маршрутизации (см. подп. 1.2.1), содержащихся в заголовке пакета сетевых адресов отправителя и получателя и служебной информации, а также текущего состояния маршрутов;

- продвижение пакетов по этим маршрутам, по цепи подсетей и маршрутизаторов (см. рис. 1.5) от отправителя получателю.

Как следует из вышесказанного, сетевой уровень имеет практический смысл в основном для составных сетей. В ВС, состоящих только из одной подсети (сегментированной или построенной в виде моноканала), сетевой уровень как таковой и средства его реализации обычно отсутствуют [3, 7]. К таким сетям относятся многие ЛВС.

Примерами протоколов сетевого уровня являются [2, 3, 7]:

- различные версии протокола IP (Internet Protocol), регламентирующего адресацию абонентов составных сетей и процесс продвижения по ним пакетов дейтаграммным способом;

- RIP (Routing Information Protocol), представляющий собой протокол дистанционно-векторной маршрутизации (см. подп. 1.2.1) с периодическим обновлением маршрутной информации;

- ICMP (Internet Control Message Protocol), регламентирующий процедуры обмена сообщениями между маршрутизаторами сети и источниками пакетов об ошибках их передачи (невозможности доставки пакета, аномальных значениях его параметров и т. п.), а также о текущем состоянии сети (например, о том, что какой-либо маршрутизатор не отвечает).

Основными средствами реализации функций сетевого уровня являются *маршрутизаторы* (см. рис. 1.5 и пояснения к нему), а также ПО сетевого уровня.

Необходимо отметить, что при реализации функций сетевого уровня не важно, по каким протоколам / стандартам реализованы подсети, через которые проходят маршруты продвижения пакетов. Обеспечение передачи пакетов через каждую из подсетей в соответствии с ее протоколами / стандартами осуществляется на канальном и физическом уровне (см. далее). Однако средства реализации сетевого уровня модели OSI, очевидно, должны быть совместимы по протоколам в пределах всей составной сети.

Основной задачей, решаемой на *канальном* и *физическом* уровнях, как указано выше, является передача данных в пределах одной подсети, между двумя ее абонентами. В их качестве могут выступать как ПК и/или серверы, так и подключенное к соответствующей подсети коммутационное оборудование (маршрутизаторы, коммутаторы и т. п.).

Протоколы / стандарты *канального* уровня и средства их реализации обеспечивают *логическое* взаимодействие абонентов подсети, в том числе [2, 3, 7]:

- оформление данных, подлежащих передаче между абонентами подсети (например, пакетов) в кадры (см. рис. 1.6), форматы которых соответствуют протоколам конкретной подсети, в том числе снабжение кадров локальными адресами отправителя и получателя;

- управление доступом абонентов к моноканалу при их запросах на передачу кадров через моноканал, а также разрешение конфликтов при одновременной попытке доступа к моноканалу двух или нескольких абонентов;

- инициализацию сеансов связи между абонентами подсети и поддержку соединений в течение этих сеансов;

- управление передачей кадров адресатам через подсеть, в том числе в сегментированных подсетях (см. рис. 1.4) – передачу кадра только в тот сегмент, в котором находится абонент-получатель;

- контроль и исправление ошибок передачи кадров, в том числе их снабжение на передающей стороне контрольными разрядами, вычисляемыми в соответствии с протоколом помехоустойчивого кодирования подсети, проверку на приемной стороне наличия ошибок в кадре (по его контрольным разрядам) и запрос повторной передачи кадра при обнаружении в нем ошибок;

- разрыв соединения между абонентами подсети по окончании сеанса связи между ними.

В свою очередь, протоколы / стандарты *физического* уровня и средства их реализации обеспечивают *физическое* взаимодействие абонентов подсети, в том числе [2, 3, 7]:

- на передающей стороне – представление кадров канального уровня электрическими или оптическими сигналами-носителями и их передачу в канал связи;

- на приемной стороне – восстановление кадров из сигнала-носителя и их передачу средствам канального уровня для дальнейшей обработки.

Также протоколы / стандарты физического уровня регламентируют тип и параметры среды передачи данных (передающей среды) подсети, например, электрического или волоконно-оптического кабеля (данные вопросы рассмотрены далее, в п. 2.3).

Функции как физического, так и канального уровней в настоящее время, как правило, реализуются посредством одного и того же сетевого оборудования – модемов, сетевых адаптеров, коммутаторов, интерфейсных блоков маршрутизаторов и т. п. (в совокупности с программным обеспечением / драйверами перечисленного оборудования) [3, 7]. При этом функции указанных двух уровней часто регламентируются одними и теми же стандартами / протоколами. Их типовыми примерами являются [3, 7]:

- стандарты группы IEEE 802.11, оговаривающие функционирование беспроводных Wi-Fi-ЛВС на физическом и канальном уровнях;

- стандарты группы IEEE 802.16, регламентирующие обмен данными по беспроводным абонентским WiMAX-окончаниям (подп. 2.2.3) на тех же уровнях;

- стандарты группы IEEE 802.3, применяемые в Ethernet-ЛВС.

Естественно, известны и стандарты / протоколы только канального или только физического уровня [1, 3, 7]. В частности, протоколы модуляции, применяемые для передачи цифровых данных по кабельным абонентским линиям телефонной сети общего пользования (V.34, V.90, V.92 и др.) относятся только к физическому уровню [1, 3, 7].

Необходимо отметить, что поддержка одних и тех же стандартов / протоколов канального и физического уровней требуется только в пределах подсети (но не сети в целом).

Взаимодействие протоколов и средств сетевого, канального и физического уровней модели OSI вкратце описано в подп. 1.2.2, в пояснениях к процессу обмена данными между абонентами составной сети. Подробное изложение вопросов реализации функций физического, канального и сетевого уровней будет представлено в гл. 2, 3 и 4 соответственно.

Краткие характеристики вышеописанных уровней модели OSI (применительно к ВС) представлены в табл. 1.3. Аббревиатуры в ней совпадают с аналогичными на рис. 1.7.

Таблица 1.3

Краткие характеристики уровней модели OSI (применительно к ВС)

Уровень	Тип данных	Основные функции	Средства реализации	Примеры протоколов
1	2	3	4	5
ПК	Сообщение	Взаимодействие приложений удаленных абонентов	Сетевое ПО	HTTP, FTP, SMTP, POP3
ПС	Поток данных	Представление и кодирование данных		NCP, NDR, XDR, JPEG, MPEG, SSL
СН	Фрагмент	Управление сеансом связи: его инициализация, поддержка и завершение		ISO-SP, NetBIOS, PPTP, RPC, RTCP
ТР	Сегмент	Управление доставкой данных между конечными пунктами маршрута		TCP, UDP, SCTP, SST, SPX

Окончание табл. 1.3

1	2	3	4	5
СТ	Пакет	Маршрутизация данных	Маршрутизаторы и их ПО	IP, RIP, ICMP, ARP
КН	Кадр	Логическое взаимодействие абонентов в пределах подсети	Сетевые адаптеры, модемы, коммутаторы, интерфейсные блоки маршрутизаторов и ПО / драйверы перечисленных устройств (блоков)	IEEE 802.3, IEEE 802.11, IEEE 802.16, HDLC, ATM
ФЗ	Сигнал – носитель данных	Физическое взаимодействие абонентов в пределах подсети		IEEE 802.3, IEEE 802.11, IEEE 802.16, V.34, V.90, V.92

Следует отметить, что кроме описанной выше модели OSI, известны и другие модели взаимодействия абонентов ВС [3]: TCP/IP, ARPANET и др. Однако модель OSI большинством специалистов считается в наибольшей степени соответствующей общему случаю реализации ВС и поэтому наиболее распространена. В частности, на модели OSI основывается изложение материала большинства монографий и учебников по тематике ВС, в том числе и дальнейшее изложение материалов настоящего учебного пособия.

Необходимо также заметить, что каждой из вышеназванных моделей взаимодействия абонентов ВС соответствует конкретный *стек* (набор) *протоколов*, каждый из которых предназначен для реализации функций определенного уровня соответствующей модели. При этом интересно отметить, что, несмотря на значительно большую популярность модели OSI, чем всех остальных моделей, при изложении принципов реализации ВС, стек протоколов OSI не нашел распространения на практике [3, 7]. С другой стороны, наиболее широко распространенным стеком протоколов является TCP/IP, в то время как модель TCP/IP сравнительно редко используется в качестве базы при изложении принципов построения ВС [2, 3, 7].

Последующие главы настоящего учебного пособия посвящены основам реализации описанных ранее уровней модели OSI.

Выводы по главе 1

ВС представляет собой совокупность объединенных некоторой системой связи территориально распределенных компьютеров, ориентированную на коллективное использование указанными компьютерами общесетевых аппаратных, программных и информационных ресурсов.

Использование ВС предоставляет ее пользователям следующие основные *преимущества*: быстрый доступ к источникам информации непосредственно с рабочего места, оперативный обмен сообщениями, документацией и денежными средствами, возможность разделения пользователями дорогостоящих аппаратно-программных общесетевых ресурсов, свободу в территориальном размещении компьютеров. Благодаря этим преимуществам, ВС в настоящее время являются одной из важнейших компонент информационных технологий, в значительной степени определяющей технологический облик современного общества.

Основным компонентом ВС является система связи, представляющая собой совокупность аппаратных и программных средств, обеспечивающих обмен данными между абонентами ВС, определяющая эффективность обмена данными между абонентами ВС и, как следствие, эффективность работы ВС в целом. Поэтому основной проблемой теории и практики ВС является создание протоколов, алгоритмов, аппаратных и программных средств связи между абонентами ВС.

Наиболее распространенным критерием *классификации* ВС является территориальный. По нему выделяют следующие основные типы ВС: локальные (ЛВС, LAN) и глобальные (ГВС, WAN). ЛВС характеризуются относительно малым числом абонентов (порядка нескольких десятков – сотен), распределенных по сравнительно небольшой территории (не более нескольких километров в радиусе), а также специально выделенными для ЛВС высококачественными, скоростными линиями связи. ГВС, в свою очередь, характеризуются достаточно большим числом абонентов (от десятков тысяч до сотен миллионов), территориально распределенных по различным городам, странам и континентам, а также использованием как специально выделенных каналов связи, так и каналов связи общего пользования при отсутствии гарантии высокого качества обмена данными. При этом ГВС обычно представляет собой составную сеть, т. е. совокупность территориально распределенных и, в общем случае, разнородных ЛВС, объединенных между собой системой связи. Кроме ЛВС и ГВС, выделяют также ряд промежуточных типов ВС [3, 7], в частности, сети мегаполисов (MAN). В настоящее время наблюдается тенденция к сближению принципов реализации ЛВС и ГВС.

Основными *характеристиками* ВС являются: производительность, надежность, информационная безопасность, расширяемость, масштабируемость, прозрачность, поддерживаемые сетью виды трафика, управляемость, а также совместимость (интегрируемость).

При *построении* ВС любого типа и назначения необходимо решать следующие основные *проблемы* [3, 7]:

- организацию надежного и высокопроизводительного обмена данными между абонентами ВС по линиям связи;
- выбор рациональной структуры ВС, топологии физических и логических связей между ее абонентами;
- создание системы адресации абонентов ВС, обеспечивающей их уникальную идентификацию;
- обеспечение совместной работы в составе ВС разнотипных и разнородных аппаратных и программных средств;
- организацию «прозрачного» (т. е. независимого с точки зрения пользователя от состава, физической и логической топологии ВС) доступа абонентов ВС к общесетевым ресурсам (информационным, аппаратным или программным).

Обмен данными между абонентами ВС, в общем случае, предполагает решение следующих *задач*:

- компактное и помехоустойчивое кодирование подлежащих передаче данных и, при необходимости, их защита от несанкционированного доступа;
- представление подвергнутых кодированию данных в формате, обеспечивающем их доставку абоненту, которому они предназначены, и их корректное распознавание получателем;
- представление данных сигналами-носителями с параметрами и характеристиками, предпочтительными для передачи данных по тому или иному типу канала связи;
- извлечение (детектирование) данных, представляемых сигналом-носителем, из этого сигнала, их корректное распознавание и декодирование на приемной стороне.
- установление и разрыв соединения между абонентами;
- коммутация и маршрутизация данных.

Решение вышеперечисленных задач осуществляется посредством сетевого оборудования (модемов, сетевых адаптеров, коммутаторов, маршрутизаторов и т. п.), в том числе его ПО, а также сетевого ПО абонентских компьютеров.

Установление соединения между абонентами ВС, *коммутация и маршрутизация данных* могут осуществляться одним из двух основных методов: *коммутации каналов* и *коммутации пакетов* (см. рис. 1.2 и пояснения к нему). На практике более распространен метод коммутации пакетов; коммутация каналов применяется в основном при передаче мультимедийного трафика в реальном масштабе времени. Основными способами передачи пакетов по сети являются: дейтаграммный, с формированием логических соединений и с формированием виртуальных каналов. На практике применяются все три перечисленных способа. Выбор маршрутов передачи пакетов (их маршрутизация) осуществляется по специальным алгоритмам (адаптивной маршрутизации; маршрутизации от источника; лавинной маршрутизации; маршрутизации, управляемой событиями). Исходными данными при маршрутизации служат адреса отправителя и получателя пакета, а также указываемые в его заголовке требования к качеству доставки (максимальная надежность, минимальная задержка и т. п.).

Структура ВС, конфигурация связей между ее абонентами и система их адресации определяются, в первую очередь, территорией, охватываемой ВС, и количеством ее абонентов. Известны следующие базовые структурные решения ВС: моноканал, сегментированная коммутируемая сеть, составная сеть (см. рис. 1.3 – 1.5 и пояснения к ним). В виде моноканала реализуются в основном достаточно простые ЛВС, с количеством абонентов не более нескольких десятков. Большинство ЛВС строится в виде сегментированных коммутируемых сетей. ГВС и достаточно крупные ЛВС (с числом абонентов более нескольких сотен и радиусом охватываемой территории порядка нескольких километров) обычно реализуются в виде составных сетей.

Обеспечение совместной работы в составе ВС разнотипных и разнородных аппаратных и программных средств и организация «прозрачного» доступа абонентов ВС к общесетевым ресурсам осуществляются за счет реализации указанных средств по принципу *открытых систем*, т. е. в соответствии с общедоступными стандартами, протоколами и рекомендациями, едиными в пределах сети или подсети. Их группирование и упорядочивание осуществляются на основе иерархических многоуровневых *моделей взаимодействия* абонентов ВС в процессе обмена данными. Наиболее популярной из них является базовая эталонная модель взаимодействия открытых систем,

обычно называемая моделью OSI (см. рис. 1.7 и пояснения к нему). На данной модели основывается дальнейшее изложение материалов настоящего учебного пособия.

Вопросы для самопроверки

1. Что является основным назначением ВС?
2. Приведите примеры общесетевых аппаратных, программных и информационных ресурсов.
3. Что является основным компонентом ВС?
4. В чем состоит основное различие между ЛВС и ГВС?
5. Дайте определения основных параметров ВС, характеризующих ее производительность: времени реакции, пропускной способности, задержки передачи.
6. Какими параметрами характеризуется надежность ВС?
7. Поясните различие между расширяемостью и масштабируемостью ВС.
8. В чем состоит прозрачность ВС на уровне пользователя и на уровне программиста?
9. Охарактеризуйте особенности компьютерного и мультимедийного трафиков.
10. Перечислите основные проблемы реализации ВС.
11. Поясните сущность методов коммутации каналов и коммутации пакетов.
12. В чем состоит сущность дейтаграммного способа передачи пакетов?
13. Поясните различие между способами передачи пакетов с формированием логических соединений и с формированием виртуальных каналов.
14. По каким причинам коммутация пакетов более распространена в практике ВС, чем коммутация каналов? В каких случаях рационально применять коммутацию каналов?
15. Дайте определение моноканала. Обоснуйте, по каким причинам в настоящее время наиболее распространена шинная топология моноканала.
16. Какие категории ВС рационально реализовывать в виде моноканала?

17. Опишите принцип построения, перечислите основные достоинства и недостатки сегментированных коммутируемых ВС.

18. Поясните принцип реализации и функционирования составной сети.

19. В чем заключается различие между локальным и сетевым адресами? Каково происхождение второго названия локального адреса – «аппаратный»?

20. Поясните смысл понятий «кадр» и «пакет», каково назначение основных полей кадра и пакета (см. рис. 1.6)?

21. Разъясните принцип открытой системы применительно к ВС.

22. Перечислите уровни модели OSI и основные функции, решаемые на этих уровнях.

23. Поясните различия между задачами, решаемыми на сеансовом и транспортном уровнях.

24. По какой причине уровни модели OSI с прикладного по транспортный включительно называют сквозными?

25. Обоснуйте отсутствие средств реализации сетевого уровня модели OSI в большинстве ЛВС.

26. Существует ли необходимость единства стандартов / протоколов канального и физического уровня в пределах составной сети? В пределах подсети? Ответ обоснуйте.

27. Поясните различия между понятиями «модель взаимодействия абонентов» и «стек протоколов ВС».

2. СЕТЕВЫЕ ТЕХНОЛОГИИ ФИЗИЧЕСКОГО УРОВНЯ

Для передачи сообщений между абонентами ВС используется, в общем случае, совокупность *физических каналов связи* (*ФКС*), последовательно соединенных между собой посредством различных устройств *коммутации* и *мультиплексирования*. *ФКС* обычно определяется как *фрагмент коммуникационной системы ВС, обеспечивающий обмен информацией между 2-мя соседними узлами ВС и не содержащий промежуточных устройств коммутации или мультиплексирования* [3]. ФКС известны в литературе также под названием *звеньев* (в англоязычной литературе – *links*) коммуникационной системы ВС [2, 7].

Основной функцией физического уровня модели OSI, как указано ранее, является обеспечение обмена потоками битов между 2-мя узлами ВС по ФКС. Данная функция сводится к реализации следующих базовых процедур:

- преобразованию кадра данных, подлежащего передаче между абонентами некоторого ФКС, в сигнал, пригодный для передачи по соответствующему ФКС;
- обратному преобразованию указанного сигнала в исходный кадр данных на приемной стороне.

При этом на физическом уровне данные, подлежащие передаче по ФКС, представляются как однородный поток битов, безотносительно к формату и смысловому назначению отдельных групп битов вышеназванного кадра. Их логическая интерпретация относится к канальному уровню, методы и средства реализации функций которого будут рассмотрены далее, в гл. 3.

Настоящая глава посвящается основам реализации функций физического уровня.

2.1. Общие сведения о ФКС ВС

2.1.1. Обобщенная структурная схема ФКС

Базовая структурная схема ФКС представлена на рис. 2.1.

ФКС обеспечивает связь типа «точка-точка» между двумя единицами *оконечного оборудования данных* (*ООД*), по-англ. – *Data*

Terminal Equipment (*DTE*). К ООД относят устройства, вырабатывающие данные, подлежащие передаче по ФКС. В качестве ООД может выступать или абонентский компьютер, или некоторый узел коммуникационной системы ВС, например, коммутатор ЛВС, концентратор коммутируемой телефонной сети общего пользования (КТСОП) и т. п. Как правило, собственно ООД не включают в состав ФКС.

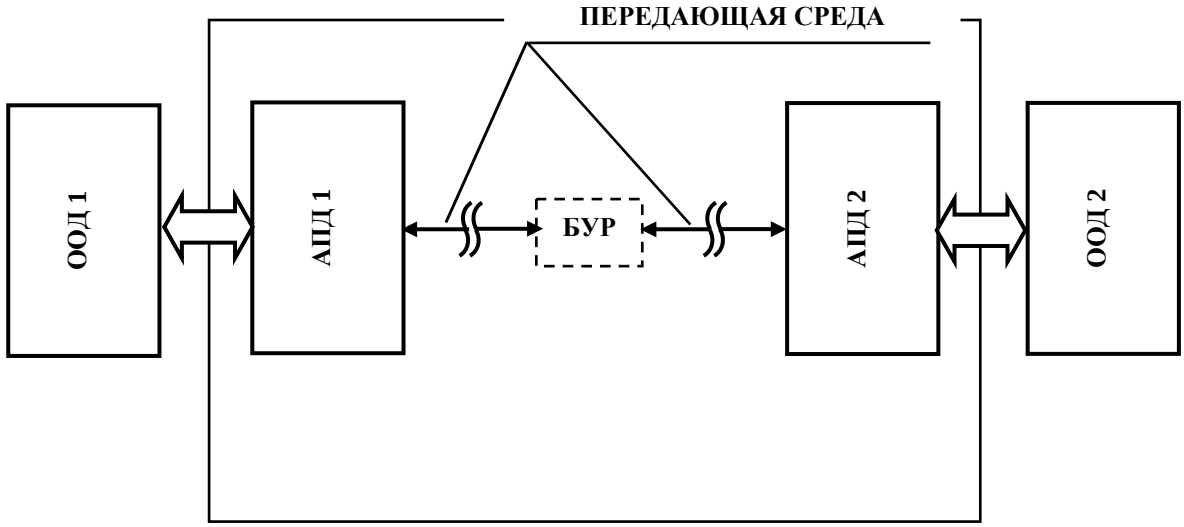


Рис. 2.1. Обобщенная структурная схема ФКС ВС:
ООД – оконечное оборудование данных; АПД – аппаратура передачи данных;
БУР – блок усиления и регенерации

Для обмена данными между ООД служит *физическая среда передачи данных* (в дальнейшем, для краткости – *передающая среда*), в качестве которой может выступать провод или кабель различных типов, воздух или вакуум [1, 3, 7].

Связь ООД с передающей средой осуществляется посредством *аппаратуры передачи данных* (*АПД*), в англоязычной литературе – *Data Communication Equipment* (*DCE*). Основными функциями АПД являются:

- преобразование подлежащих передаче данных, поступающих с ООД, в сигнал, с одной стороны, несущий информацию о них, а с другой – пригодный для обмена данными по передающей среде соответствующего типа;
- преобразование сигнала-носителя, принимаемого из передающей среды, в данные, поступающие на ООД.

В качестве АПД могут выступать, например, *модуляторы / демодуляторы (модемы)* и/или *сетевые адаптеры* абонентских компьютеров, а также аналогичные им блоки коммутационного оборудования ВС. Необходимо отметить, что в настоящее время АПД часто не является отдельным конструктивным узлом, а входит в состав ООД [3].

Блоки усиления и регенерации (*БУР*) являются *промежуточной аппаратурой* ФКС. Они выполняют функции повышения мощности сигнала – носителя данных и восстановления его формы при искажениях, вносимых в процессе передачи сигнала. Их применение необходимо при протяженности ФКС больше некоторого значения, определяемого конкретным типом передающей среды, параметрами сигнала – носителя данных и некоторыми другими факторами [1, 3]. Например, при использовании кабеля на основе неэкранированной витой пары проводов [3] в качестве передающей среды, сигнале-носителе типа «Манчестер» (описанного далее, в подп. 2.5.5) и скорости обмена данными 10 Мбит/с максимальная длина кабеля, не требующая применения БУР, равна 100 м [3].

2.1.2. Сигналы-носители информации в ФКС

В качестве данных сигналов выступают электрические напряжения, токи или электромагнитные волны различных частотных диапазонов, в зависимости от конкретного типа ФКС. Информацию о передаваемых данных несут изменения во времени состояния сигнала-носителя, которое характеризуется значениями одного или нескольких *информативных параметров* указанного сигнала (амплитуды, частоты, фазы и т. п.).

По числу состояний сигнала-носителя, распознаваемых приемником информации, различают две формы ее представления – *непрерывную* и *дискретную*. Непрерывная форма (называемая также *аналоговой*) предполагает применение теоретически бесконечного числа указанных состояний. На практике, однако, оно ограничено отношением $\Delta X / \Delta x_{\min}$, где ΔX – допустимый диапазон изменения информативного параметра сигнала, $\Delta x_{\min}$ – минимальная распознаваемая приемником разность между двумя его значениями, определяемая

техническими возможностями приемной аппаратуры, а также уровнем шумов и помех на линии связи. При *дискретной* форме представления данных, в свою очередь, применяется *конечное* число состояний сигнала-носителя, различаемых приемником информации. Распространенным частным случаем дискретной формы представления информации является применяемое в цифровой электронике *двоичное* представление данных с двумя распознаваемыми приемником состояниями сигнала-носителя данных, информативным параметром которого служит уровень напряжения или (реже) тока. В частности, цифровые КМОП-ИС воспринимают напряжения, находящиеся в диапазоне от 0 до $(1/3)\times U_{пит}$ (где $U_{пит}$ – напряжение питания) как уровень логического нуля, а в диапазоне от $(2/3)\times U_{пит}$ до $U_{пит}$ – как уровень логической единицы. Потенциальный вариант интерфейса RS-232C воспринимает как уровень нуля напряжения в диапазоне от 5 до 15 В, а единицы – от минус 5 до минус 15 В и т. п.

Благодаря ограниченному количеству различаемых состояний сигнала-носителя дискретных данных, диапазоны значений его информативных параметров, соответствующие каждому из указанных состояний, достаточно широки (см. выше) и, как правило, существенно превышают уровень шумов и помех на линии связи. Поэтому дискретная форма представления информации отличается значительно более высокой помехоустойчивостью, чем непрерывная. Кроме того, как было указано ранее, абонентами ВС являются компьютеры или (значительно реже) терминалы различного класса, представление и обработка данных в которых осуществляется исключительно в двоичной (т. е. дискретной) форме. С учетом этих двух факторов, представление и передача данных в ФКС ВС осуществляются только в дискретной форме, с использованием передаваемой абонентом двоичной (цифровой) последовательности в качестве исходной при формировании сигнала-носителя.

На первый взгляд, естественно было бы использовать два различаемых состояния сигнала-носителя ФКС, каждое из которых соответствует определенному значению («0» или «1») очередного бита исходной двоичной последовательности. Ряд протоколов передачи данных по ФКС ВС, преимущественно более ранней разработки, предполагает именно такой подход. Однако многие типы ФКС, например, линии связи ЛВС или высококачественные телефонные линии, обла-

дая относительно низким уровнем шумов и помех, допускают дискретное представление данных с числом различаемых состояний сигнала-носителя больше двух [1, 5]. При этом каждое из них соответствует определенному сочетанию битов передаваемой двоичной последовательности.

В подавляющем большинстве практических случаев в качестве носителей двоичных данных ФКС с двумя или более различаемыми состояниями выступает один из следующих типов сигналов [1, 5]:

- двухуровневый;
- многоуровневый;
- *модулированный* синусоидальный сигнал, представляющий собой синусоиду определенной частоты (*несущую*), значения одного из параметров которой (амплитуда, частота или фаза, в ряде случаев – амплитуда и фаза в совокупности) несут информацию о передаваемом сообщении.

Примеры временных диаграмм сигналов каждого из перечисленных типов представлены на рис. 2.2. Преобразование исходных двоичных данных в указанные сигналы осуществляется посредством АПД (см. рис. 2.1). Его принципы будут рассмотрены далее (пп. 2.5 и 2.6).

При этом, естественно, передача двоичных данных по ФКС посредством сигнала-носителя каждого из ранее перечисленных типов осуществляется *последовательно*, бит за битом (или группа битов за группой).

Необходимо отметить, что во многих практических случаях одна и та же ФКС может использоваться для передачи сигналов всех вышеназванных типов. Поэтому должны существовать *универсальные характеристики* сигналов, позволяющие определять параметры выходного сигнала ФКС при любом типе ее входного сигнала. Наиболее распространенной из таких характеристик является *спектральный состав* (*спектр*) сигнала, т. е. результат его представления в виде суммы некоторых *базисных функций*. Они определяются как набор функций, удовлетворяющих требованиям, в упрощенной форме выражаемых следующим образом [5]:

- сигнал любой формы может быть представлен в виде конечной или бесконечной суммы базисных функций;
- сигнал может быть однозначно восстановлен из данной суммы с любой заданной точностью.

До настоящего времени наиболее распространенными базисными функциями являются синусоидальные (гармонические) [5]. При этом спектральное представление сигнала имеет следующий вид:

$$X(t) = \sum_{i=0}^{n} A_i \times \sin\left(2\pi \times f_i \times t + \varphi_i\right), \qquad (2.1)$$

где A_i, f_i, φ_i – соответственно амплитуда, частота и начальная фаза i-й синусоидальной компоненты (составляющей) сигнала, называемой также *спектральной компонентой*, *спектральной составляющей*, или *гармоникой*.

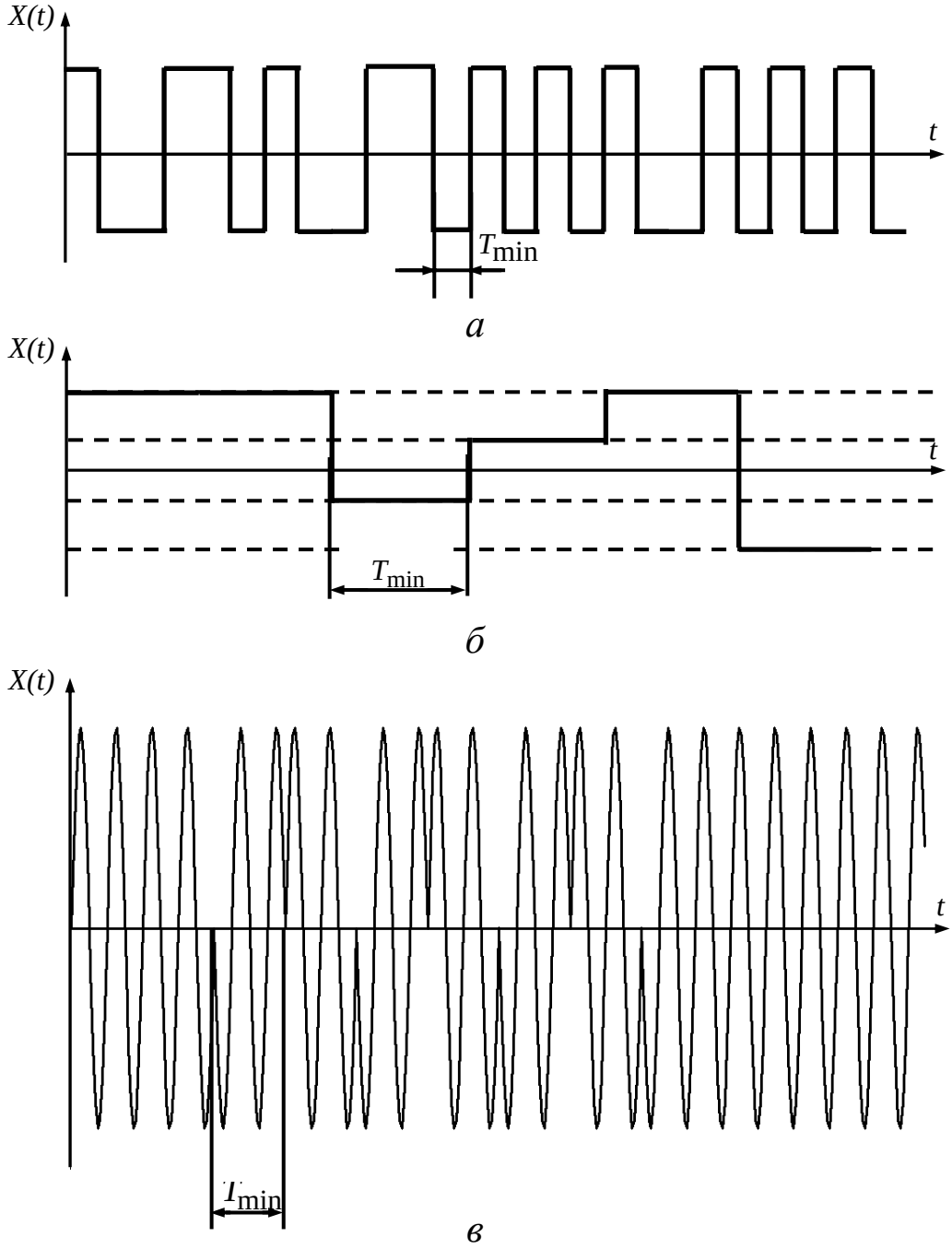


Рис. 2.2. Примеры временных диаграмм сигналов-носителей данных ФКС ВС: *а* – двухуровневого; *б* – четырехуровневого; *в* – синусоиды, модулированной по фазе

Дальнейшее изложение основ передачи данных по ФКС ВС будет основываться на представлении сигнала-носителя информации в гармоническом базисе, в соответствии с выражением (2.1).

Число n спектральных компонент теоретически бесконечно. Однако на практике спектральное представление сигнала всегда осуществляется *с некоторой заданной погрешностью*, отличной от нуля. При таком представлении в любом случае можно ограничиться некоторым конечным числом спектральных компонент [5].

Под собственно *спектром* сигнала обычно подразумевается зависимость $A_i(f_i)$ (амплитудный спектр) и/или $\varphi_i(f_i)$ (фазовый спектр) [5] (см. выражение (2.1)). На практике обычно более информативен амплитудный спектр [5]. Из его характеристик при передаче сигналов по каналам связи, в том числе по ФКС ВС, наиболее важное практическое значение имеют *граничные частоты амплитудного спектра сигнала*. Они определяются как границы диапазона частот, в котором находятся *значимые* спектральные составляющие сигнала. К ним относят компоненты с амплитудами, удовлетворяющими условию:

$$A_i / A_{\max} \geq \delta_{\min}, \qquad (2.2)$$

где $A_{\max}$ – максимальная из амплитуд спектральных компонент сигнала;

$\delta_{\min}$ – минимальное отношение амплитуды спектральной составляющей сигнала к $A_{\max}$, при котором она считается значимой [5] (т. е. ею нельзя пренебречь).

В различных практических приложениях передачи и обработки сигналов используются различные значения $\delta_{\min}$, в зависимости от требований к точности представления и восстановления сигнала, предъявляемых конкретным приложением. При передаче дискретных данных по линиям связи, в том числе по ФКС ВС, значение $\delta_{\min}$ обычно принимается равным 0,1 или 0,3 [1, 5], в то время как, например, в информационно-измерительной технике оно может задаваться на уровне порядка нескольких тысячных и менее.

Распространенные разновидности сигналов-носителей двоичных данных в ФКС ВС (см. рис. 2.2), в общем случае (т. е. при равной вероятности всех возможных двоичных комбинаций исходных данных),

характеризуются следующими граничными частотами амплитудного спектра при $\delta_{\min}$, равном 0,3 [1, 5]:

- двух- и многоуровневый сигнал-носитель (см. рис. 2.2, *а* и 2.2, *б*):

$$\left.\begin{array}{r} f_{LS} = 0; \\ f_{HS} \approx 1/T_{\min}; \end{array}\right\}; \tag{2.3}$$

- модулированный синусоидальный сигнал-носитель (см. рис. 2.2, *в*):

$$\left.\begin{array}{l} f_{LS} \approx f_0 - (1/T_{\min}); \\ f_{HS} \approx f_0 + (1/T_{\min}); \end{array}\right\}, \tag{2.4}$$

где f_{LS} и f_{HS} – соответственно нижняя и верхняя граничная частота амплитудного спектра сигнала;

$T_{\min}$ – минимальная длительность интервала времени между изменениями информативных параметров сигнала-носителя (см. рис. 2.2);

f_0 – частота несущей.

Вследствие этого двух- и многоуровневые сигналы-носители в основном применяются в ФКС, полоса пропускания которых (подп. 2.1.3) включает в себя нулевую частоту или частоты, близкие к нулевой, например, в кабельных каналах связи ЛВС. Модулированные синусоидальные сигналы-носители, в свою очередь, применяются в ФКС, нижняя граничная частота полосы пропускания которых существенно отлична от нуля, например, в беспроводных ФКС (подп. 2.2.3).

2.1.3. Основные параметры и характеристики ФКС

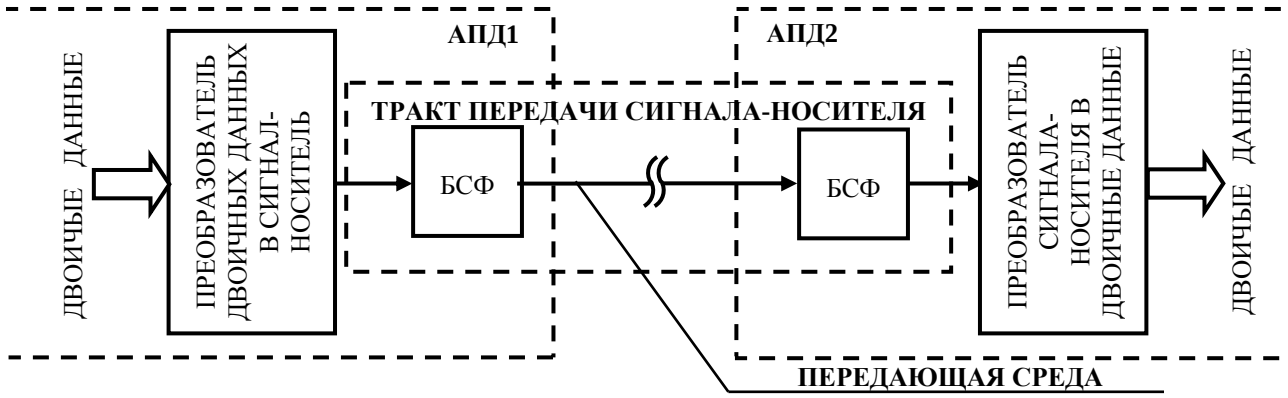
К базовым характеристикам *ФКС в целом* относятся [1, 3, 5]:

- полоса пропускания (*bandwidth*);
- пропускная способность (*throughput*);
- достоверность передачи данных (*bit error rate, BER*).

Полоса пропускания ФКС на практике обычно определяется как *диапазон частот, в пределах которого спад амплитудно-частотной характеристики (АЧХ) тракта передачи сигнала-носителя данных ФКС не превышает некоторого заданного значения,* $\delta_{\max}$.

Под *трактом передачи сигнала-носителя данных* подразумевается участок ФКС от выхода преобразователя передаваемых двоичных данных в сигнал-носитель до входа преобразователя этого сигнала

в исходные двоичные данные. Указанные преобразователи, в свою очередь, обычно входят в состав АПД [1, 3, 5]. Обобщенная структурная схема указанного тракта (с источником и приемником сигнала-носителя) представлена на рис. 2.3. Необходимо отметить, что на практике связь между абонентами ФКС является двусторонней. Во избежание загромождения, на рис. 2.3 не показаны элементы тракта передачи данных от АПД2 к АПД1.



БСФ – блоки согласования и фильтрации

Рис. 2.3. Обобщенная структурная схема тракта передачи данных в ФКС ВС

АЧХ ФКС, обозначаемая символом $|H(f)|$, обычно определяется как зависимость от частоты следующего отношения:

$$|H(f)| = X_{mo}(f)/X_{mi}(f), \qquad (2.5)$$

где $X_{mi}(f)$ и $X_{mo}(f)$ – амплитуда синусоидального сигнала частотой f соответственно на входе и на выходе тракта передачи сигнала-носителя (см. рис. 2.3) при некоторых заданных значениях выходного сопротивления БСФ передающего абонента и входного сопротивления БСФ абонента, принимающего данные [1, 5].

Из рис. 2.3 можно увидеть, что АЧХ ФКС и, соответственно, ее полоса пропускания определяются АЧХ (полосой пропускания) передающей среды, а также БСФ абонентских АПД. При этом необходимо отметить, что АЧХ передающей среды описывается выражением, несколько отличающимся от (2.5) (см. представленные далее определения параметров передающей среды ФКС).

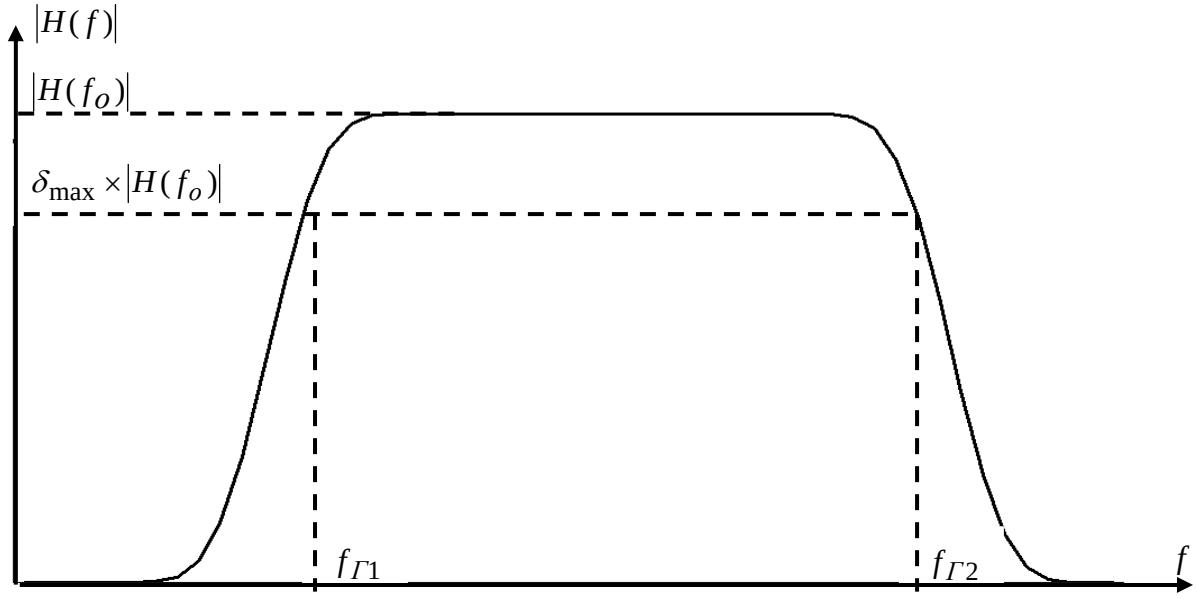
Спад АЧХ ФКС на некоторой частоте f, которую обозначим $\delta(f)$, определяется как отношение:

$$\delta(f) = \frac{|H(f_o)| - |H(f)|}{|H(f_o)|}, \qquad (2.6)$$

где f_o – некоторая частота в пределах полосы пропускания ФКС, значение АЧХ на которой принимается за опорное. Если полоса пропускания включает в себя нулевую частоту, она выступает в качестве f_o. В противном случае, f_o, как правило, выбирается равной центральной частоте полосы пропускания [1, 5].

Значение максимального спада АЧХ ФКС $\delta_{\max}$, определяющего границы полосы пропускания, обычно задается равным 0,5 или 0,3 [1].

Смысл понятия полосы пропускания поясняет рис. 2.4.



$f_{Г1}$, $f_{Г2}$ – граничные частоты полосы пропускания

Рис. 2.4. Пояснение смысла понятия «полоса пропускания»

От ширины полосы пропускания ФКС зависит второй из перечисленных ранее базовых параметров ФКС – *пропускная способность*. Она определяется как *максимально возможная скорость передачи двоичных данных по ФКС, выражаемая в битах в секунду*.

Зависимость пропускной способности ФКС от ширины его полосы пропускания объясняется следующим. Как указано ранее, информация о передаваемых данных в ФКС ВС представляется состояниями

сигнала-носителя. Следовательно, чем выше скорость передачи данных, тем более высокая частота переключения состояний сигнала-носителя требуется для обеспечения соответствующей скорости и тем более широк частотный диапазон указанного сигнала [1, 5].

В наиболее общем виде связь между шириной полосы пропускания ФКС и ее пропускной способностью выражается *формулой Шеннона* [1, 3, 5], имеющей следующий вид:

$$C = \Delta f \times \log_2 \{1 + (P_C / P_{Ш})\}, \qquad (2.7)$$

где Δf – ширина полосы пропускания ФКС;

P_C и $P_{Ш}$ – соответственно мощность сигнала-носителя и шума на ФКС.

Выражение (2.7) предполагает, что число различаемых приемником состояний сигнала-носителя определяется отношением $P_C / P_{Ш}$ на линии. При $P_{Ш} = 0$ количество указанных состояний теоретически может быть бесконечным; бесконечна при этом и пропускная способность ФКС (что, однако, невозможно на практике).

Более удобной для практического применения является *формула Найквиста* [1, 3, 5]:

$$C = 2 \times \Delta f \times \log_2 N, \qquad (2.8)$$

где N – число различаемых приемником состояний сигнала-носителя при используемом методе представления данных в передающей среде ФКС.

Согласно выражению (2.8), как и следует ожидать, пропускная способность ФКС прямо пропорциональна ширине ее полосы пропускания, а также числу битов, кодируемому каждым из различаемых состояний сигнала-носителя и равному $\log_2 N$.

Необходимо отметить, что выражения (2.7) и (2.8) определяют *максимально возможную* скорость передачи данных по ФКС при заданных ширине ее полосы пропускания и числе различаемых состояний сигнала-носителя или, соответственно, отношении «сигнал-шум». Реально достижимая скорость передачи всегда меньше максимально возможной и определяется, кроме вышеназванных параметров и характеристик ФКС, также типом сигнала-носителя и методом преобразования передаваемых двоичных данных в его информативные параметры. Связь между полосой пропускания ФКС и реально достижи-

мой скоростью передачи данных для распространенных методов указанного преобразования будет устанавливаться при конкретном рассмотрении каждого из них (пп. 2.5 и 2.6).

Достоверность передачи данных определяется как *вероятность искажения каждого из передаваемых по ФКС битов данных*. Этот параметр ФКС известен также под названием «*интенсивность битовых ошибок*», в англоязычной литературе – *Bit Error Rate* (*BER*). BER, естественно, всегда меньше или равна единице, причем ее равенство некоторому значению p соответствует тому, что в среднем один бит из $1/p$ искажается после передачи по ФКС. Значение BER зависит в основном от типа и параметров передающей среды ФКС (см. рис. 2.1), а также от применяемых методов защиты от ошибок на канальном уровне, например, помехоустойчивого кодирования передаваемых двоичных данных (подп. 2.7.4 и п. 3.8). При отсутствии такой защиты BER составляет, как правило, 10^{-3} – 10^{-6}, а при использовании оптоволоконных линий связи в качестве передающей среды – порядка 10^{-9} [3, 7]. При этом достоверность передачи данных, равная, например, 10^{-4} соответствует искажению в среднем одного бита из 10 000 при их передаче по ФКС.

Кроме рассмотренных выше параметров ФКС в целом, важное практическое значение имеют *параметры и характеристики передающей среды ФКС*, основными из которых являются [1, 3, 7]:

- полоса пропускания (*bandwidth*);
- затухание (*attenuation*);
- помехоустойчивость;
- волновое сопротивление (для электрического кабеля).

Полоса пропускания передающей среды ФКС определяется аналогично полосе пропускания ФКС в целом (см. выше), за исключением того, что вместо АЧХ при этом используется коэффициент передачи по мощности, определяемый следующим образом [3]:

$$A_p(f) = P_o(f)/P_i(f), \qquad (2.9)$$

где $P_i(f)$ и $P_o(f)$ – мощность синусоидального сигнала частотой f соответственно на входе и на выходе участка передающей среды определенной длины, а для электрического кабеля – также при заданных значениях выходного сопротивления источника сигнала и сопротивления нагрузки, обычно равных *волновому сопротивлению* кабеля (см. далее). Также следует отметить, что значение максимального

спада коэффициента передающей среды, определяющее границы полосы пропускания (см. рис. 2.4), как правило, задается равным 0,5 [3, 7].

Полоса пропускания электрического и волоконно-оптического кабеля определяется его типом, длиной и конструктивными параметрами (диаметром, шагом скрутки и т. п.) [3, 7]. Типовые значения ширины полосы пропускания для распространенных разновидностей кабеля будут представлены далее (п. 2.3). Полоса пропускания беспроводной передающей среды (например, радиоэфира) определяется длиной ее участка а также ее физико-химическими параметрами (температурой, влажностью и т. п.) [3, 7]. Для беспроводной передающей среды обычно нормируется не ширина полосы пропускания, а только *затухание* на определенных частотах и при заданных условиях [3, 7].

Затухание обычно определяется как *относительное уменьшение мощности синусоидального сигнала некоторой заданной частоты при его передаче по участку передающей среды определенной длины* [3]. Затухание электрического кабеля нормируется при выходном сопротивлении источника сигнала и входном сопротивлении приемника, равных волновому сопротивлению кабеля. Как правило, затухание выражается в *децибелах* в соответствии со следующим выражением:

$$A = 10 \times \lg\{P_o(f_T)/P_i(f_T)\}, \qquad (2.10)$$

где $P_i(f_T)$ и $P_o(f_T)$ – мощность синусоидального сигнала с заданной («тестовой») частотой f_T соответственно на входе и на выходе участка передающей среды.

Например, уменьшение мощности в 10 раз соответствует затуханию, равному минус 10 дБ, в 30 раз – минус 14,8 дБ, в 100 раз – минус 20 дБ и т. д. Отсутствие ослабления сигнала в передающей среде (практически никогда не встречающееся) соответствовало бы затуханию, равному 0 дБ.

Затухание, по существу, является выраженным в децибелах значением $A_p(f)$ отрезка передающей среды на определенной частоте. Как правило, затухание нормируется на нескольких «тестовых» частотах, что более удобно с технической и организационной точек зрения, чем нормирование $A_p(f)$ в целом. На практике важно знать затухание на частоте *основной гармоники* сигнала-носителя, т. е. его спектральной компоненты с максимальной амплитудой.

Затухание определяется теми же факторами, что и полоса пропускания (см. выше). Типовые значения затухания для различных разновидностей передающей среды будут представлены далее (п. 2.3).

Помехоустойчивость передающей среды характеризует ее способность ослаблять влияние внешних электромагнитных наводок на передаваемый сигнал [3]. Очевидно, понятие помехоустойчивости имеет смысл только для кабеля. Беспроводная передающая среда практически не обладает устойчивостью к помехам, вследствие чего их подавление/устранение в беспроводных ФКС осуществляется исключительно посредством блоков согласования и фильтрации (см. рис. 2.1).

Основными *количественными характеристиками* помехоустойчивости являются [3]:

- перекрестные наводки на ближнем конце (*Near End Cross Talk, NEXT*);
- перекрестные наводки на дальнем конце (*Far End Cross Talk, FEXT*).

Смысл параметров NEXT и FEXT поясняет нижеприведенный рис. 2.5.

NEXT характеризует влияние выходного синусоидального сигнала передатчика, подключенного к одному из кабелей некоторого многокабельного ФКС, на вход приемника, подключенного к соседнему кабелю того же ФКС и находящегося в составе той же АПД, что и вышеуказанный передатчик. NEXT выражается в децибелах и определяется в соответствии со следующим выражением [3]:

$$NEXT = 10 \times \lg\{P_{IR}(f_T)/P_{OT}(f_T)\}, \qquad (2.11)$$

где $P_{OT}(f_T)$ и $P_{IR}(f_T)$ – соответственно мощность синусоидального сигнала с «тестовой» частотой f_T на выходе передатчика и мощность вызванной им наводки на входе приемника соседнего кабеля.

FEXT, в отличие от NEXT, характеризует влияние выходного сигнала передатчика одного из кабелей многокабельного ФКС на вход приемника, подключенного к соседнему кабелю того же ФКС на стороне, *противоположной* передатчику (рис. 2.5). Выражение для расчета FEXT совпадает с (2.11), за исключением того, что в качестве $P_{IR}(f_T)$ в нем выступает мощность сигнала на входе приемника противоположной стороны соседнего кабеля.

Необходимо отметить, что в ряде научно-технических источников (в т. ч. справочников) в выражении вида (2.11), определяющем NEXT и FEXT, вместо отношения $P_{IR}(f_T)/P_{OT}(f_T)$ используется обратное

ему отношение, т. е. $P_{OT}(f_T)/P_{IR}(f_T)$. В любом из данных случаев абсолютное значение NEXT или FEXT, очевидно, одинаково; от того, какое из указанных отношений используется при определении параметра NEXT или FEXT, зависит только их знак. При использовании отношения $P_{IR}(f_T)/P_{OT}(f_T)$ эти параметры отрицательны, а в противном случае – положительны.

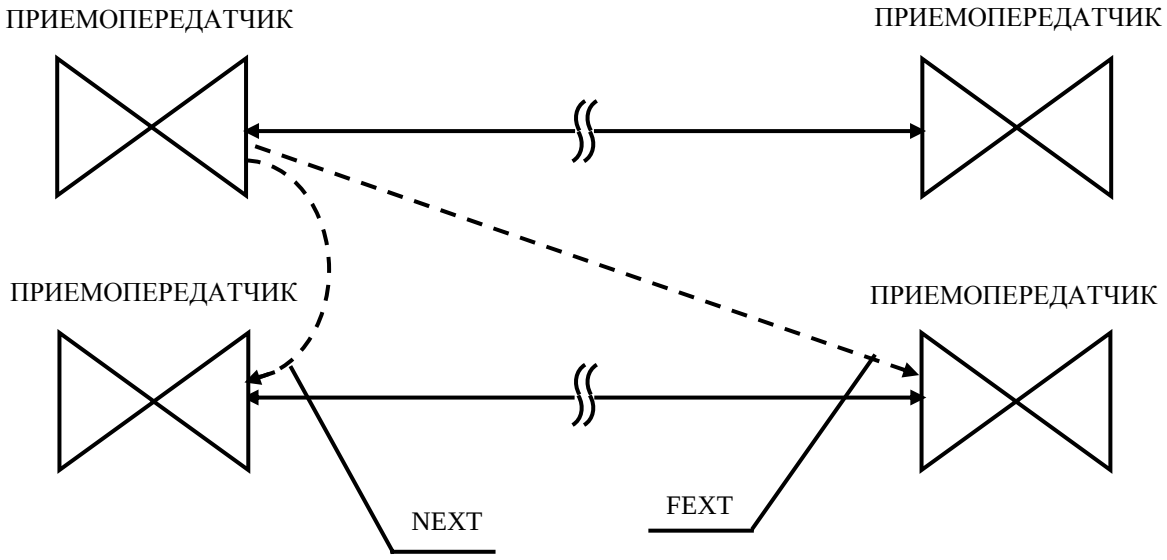


Рис. 2.5. Пояснение смысла параметров NEXT и FEXT

На практике обычно нормируется только параметр NEXT. Влияние перекрестных наводок на дальнем конце, очевидно, заведомо меньше, чем на ближнем; поэтому параметр FEXT менее информативен, чем NEXT. Как правило, NEXT указывается для нескольких «тестовых» частот. Типовые значения NEXT для распространенных разновидностей кабелей будут представлены далее (п. 2.3).

Разновидностью параметров NEXT и FEXT являются *PS-NEXT* и *PS-FEXT* (где аббревиатура соответствует английскому словосочетанию *Power Sum*, наиболее корректный перевод которого – «суммарная наводка»). Более информативным параметром является PS-NEXT (см. вышеприведенные рассуждения для NEXT и FEXT), определяемый в соответствии со следующим выражением [3]:

$$PS-NEXT = 10 \times \lg\{P_{\Sigma IR}(f_T)/P_{\Sigma OT}(f_T)\}, \qquad (2.12)$$

где $P_{\Sigma OT}(f_T)$ – суммарная мощность синусоидальных сигналов частотой f_T на выходах всех передатчиков многокабельной ФКС, входящих в состав одной из АПД;

$P_{\Sigma IR}(f_T)$ – суммарная мощность вызванной ими наводки на входе одного из приемников ФКС, входящих в состав той же АПД.

Волновое сопротивление как параметр передающей среды имеет смысл только для электрического кабеля. Оно определяется как отношение амплитуды волны напряжения к амплитуде волны тока при распространении электромагнитной волны вдоль кабеля и отсутствии ее отражения от нагрузки. Практический смысл волнового сопротивления заключается в следующем: для отсутствия искажений высокочастотных сигналов при их передаче по кабелю сопротивление его нагрузки и выходное сопротивление источника сигнала должны быть равны волновому сопротивлению кабеля. Данные условия известны под названием *условий согласования* кабеля. При этом под высокочастотным понимается сигнал, отношение длины волны значимых спектральных компонент которого (см. выражение (2.2)) к длине кабельной линии составляет 8 – 10 и менее [3]. Напомним, что длина электромагнитной волны равна отношению v/f, где f – частота, v – скорость распространения волны в соответствующей среде. Применительно к кабелю роль данной среды играет его изоляция, так как при распространении высокочастотных электромагнитных волн по кабелю они сосредотачиваются в основном в изоляции, а проводящие жилы кабеля лишь задают направление движения волн, благодаря чему они не рассеиваются, а распространяются вдоль линии [3]. Скорость распространения электромагнитной волны в среде с относительной диэлектрической проницаемостью ε равна $3\times10^8/\sqrt{\varepsilon}$ м/с [3]. Например, даже для отрезка кабеля длиной 100 м высокочастотными и, следовательно, требующими соблюдения условий согласования, являются сигналы с длиной волны значимых спектральных компонент порядка 800 – 1000 м и меньше, т. е. с их частотой порядка 30 – 40 кГц и выше. Поэтому практически все ФКС ВС, использующие электрический кабель в качестве передающей среды, требуют соблюдения вышеуказанных условий согласования.

Волновое сопротивление кабеля определяется материалом его изоляции, а также конструктивными параметрами. Его типовые значения лежат в пределах от 50 до 150 Ом [3].

Обзор типовых значений вышеописанных параметров и характеристик, как распространенных типов ФКС в целом, так и их передающей среды, будет представлен в пп. 2.2 и 2.3.

2.1.4. Классификация ФКС ВС

В качестве основных критериев классификации ФКС обычно служат их назначение и тип передающей среды [1]. По данным критериям различают следующие основные типы ФКС ВС:

- кабельные ФКС КТСОП, используемые для обмена данными между абонентами ВС;
- кабельные ФКС, специально выделенные для обмена данными между узлами ВС, в том числе кабельные ФКС ЛВС;
- беспроводные ФКС общего пользования;
- выделенные беспроводные ФКС, в том числе беспроводные ФКС ЛВС.

Рассмотрим основные отличительные особенности вышеперечисленных типов ФКС.

2.2. Обзор основных типов ФКС ВС

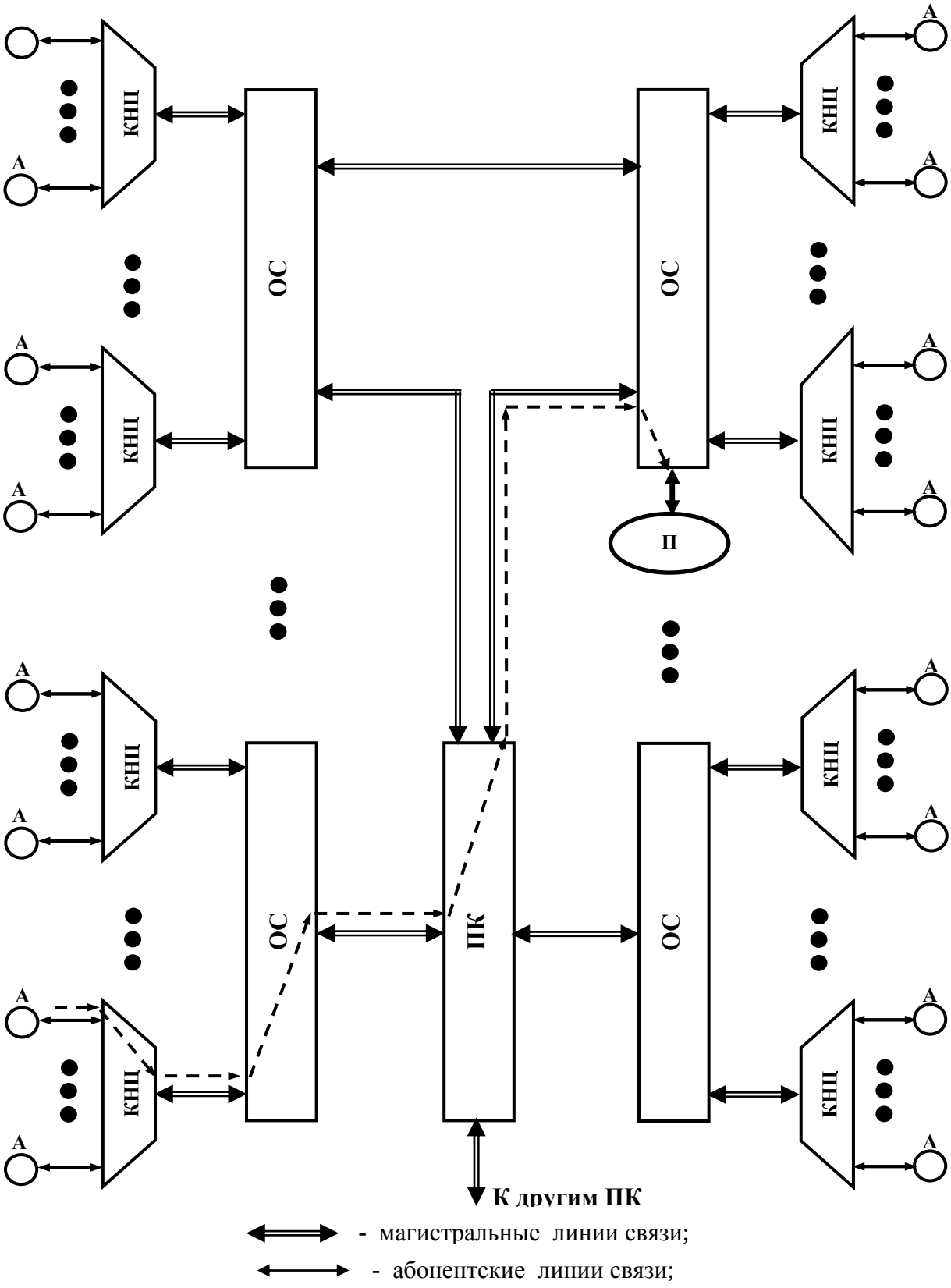
2.2.1. Кабельные ФКС КТСОП

Данный тип ФКС является наиболее давно применяемым для связи между абонентами ГВС. В самом деле, при достаточно больших расстояниях между ними и их значительном количестве (что имеет место в ГВС) для связи между абонентами экономически и организационно оправдано использовать уже существующие системы коммуникаций [3, 7]. К таковым, в первую очередь, относится КТСОП (по-англ. – *Public Switched Telephone Network, PSTN*), развивающаяся с начала прошлого века [3].

Обобщенная структурная схема фрагмента КТСОП представлена на рис. 2.6 [3, 7].

В целом, КТСОП состоит из следующих основных элементов:

- *абонентских линий связи* (АЛС);
- *магистральных линий связи* (МЛС);
- *коммутационного оборудования*, основными типами которого являются *концентраторы* (называемые также *подстанциями* или *оконечными телефонными станциями*); *опорные станции* (ОС) и *промежуточные коммутаторы* (ПК), известные также под названием *транзитных узлов*.



А – абоненты КТСОП; П – ЛВС провайдера; КНЦ – концентраторы;
ОС – опорные станции; ПК – промежуточные коммутаторы

Рис. 2.6. Обобщенная структурная схема фрагмента КТСОП

В качестве абонентского оборудования КТСОП, в общем случае, могут выступать телефонные аппараты, факсы, АПД абонентских компьютеров ВС, подключаемых к ней через КТСОП, а также коммуникационное оборудование *сетевых провайдеров*. В современных КТСОП представление данных в аналоговой форме (наряду с цифровым) используется только в АЛС, причем в АЛС провайдеров применяется в основном цифровое представление данных.

Для МЛС характерен обмен данными исключительно в цифровой форме, с использованием сигналов-носителей, аналогичных применяемым в ФКС ВС (см. подп. 2.1.2 и рис. 2.2).

Практически, во всех МЛС современных КТСОП используется *уплотнение* (*мультиплексирование*) каналов связи, т. е. применение одной и той же МЛС для организации связи между множеством абонентов КТСОП. В МЛС КТСОП используется в основном *временное* уплотнение каналов (подпункт 2.8.4).

При этом существует *организационно-законодательное ограничение* на полосу пропускания каналов связи абонентских телефонных аппаратов и факсов с оконечными телефонными станциями (концентраторами) КТСОП: под каждый из таких каналов выделяется диапазон частот от 300 до 3400 Гц [3, 7]. Данное ограничение обусловлено не физическими возможностями трактов обмена данными между абонентами и концентраторами, а, в первую очередь, необходимостью *уплотнения* каналов связи между коммутационным оборудованием КТСОП (см. выше). Уплотнение каналов, в свою очередь, требует максимально возможного ограничения спектра информативных сигналов [1, 5]. Диапазон частот от 300 до 3400 Гц был выбран как минимально необходимый для приемлемого качества обмена речевыми сигналами между абонентами КТСОП (что изначально являлось ее основным назначением) [7].

Указанное ограничение распространяется и на ФКС, соединяющие абонентские компьютеры ВС с концентраторами КТСОП, если для обмена данными с другими узлами ВС, например, с провайдером, эти компьютеры используют коммутационное оборудование КТСОП, изначально предназначенное для связи между телефонными аппаратами / факсами. Однако это ограничение устраняется, например, при использовании технологий xDSL (см. далее).

Следует отметить, что данное ограничение не распространяется на ФКС, соединяющие провайдеров с коммутационным оборудованием КТСОП.

Абоненты КТСОП через АЛС подключаются к концентраторам (оконечным станциям) КТСОП, которые выполняют следующие основные функции:

- фильтрацию сигналов, передаваемых абонентами, т. е. ограничение их спектра частотами от 300 до 3400 Гц (см. выше);
- преобразование сигналов, полученных в результате фильтрации, в цифровые коды;
- *мультиплексирование* представленных в цифровой форме сообщений, передаваемых различными абонентами, т. е. их объединение в совокупность сообщений, поступающую в МЛС, которая соединяет концентратор с ОС;
- *демультиплексирование*, т. е. извлечение цифрового кода, предназначенного каждому из абонентов, из совокупности сообщений, поступающих с ОС по МЛС на концентратор;
- преобразование указанного кода в аналоговый сигнал, его фильтрация и передача абоненту по АЛС.

Необходимо отметить, что в отдельных случаях концентраторы могут располагаться непосредственно на ОС, если это возможно технически и организационно.

ОС и ПК существенно не различаются между собой принципами реализации и представляют многопортовые «интеллектуальные» цифровые коммутаторы, распределяющие сообщения (совокупности сообщений) между МЛС в соответствии с содержащейся в сообщениях адресной информацией о пунктах назначения. ОС совместно с ПК обеспечивают формирование *составных каналов связи* между абонентами КТСОП. Под составным каналом связи подразумевается тракт обмена данными между парой абонентов, состоящий из последовательно соединенных АЛС, МЛС и внутренних соединений в коммутаторах ОС и ПК. Пример такого тракта обозначен (см. рис. 2.6) пунктирной линией. При формировании составных каналов связи в современных КТСОП используются технологии как коммутации каналов, так и коммутации пакетов (см. рис. 1.2 и пояснения к нему).

Следует отметить, что ОС и ПК современных КТСОП, как правило, снабжаются средствами сопряжения с сетями мобильной связи (вкратце описанными далее, в подп. 2.2.3), что позволяет ее абонентам осуществлять соединение с абонентами КТСОП.

Как было сказано ранее, передача данных по МЛС современных КТСОП осуществляется исключительно в цифровой форме. Ши-

рина их полосы пропускания определяется, в первую очередь, *уровнем иерархии* МЛС, т. е. количеством абонентов, использующих ее для связи, и составляет от нескольких мегагерц до нескольких сотен мегагерц, при пропускной способности от единиц до сотен Мбит/с [3].

В целом, кабельные ФКС КТСОП реализуются в соответствии со структурной схемой, приведенной на рис. 2.1, а их тракты передачи сигнала-носителя данных со структурной схемой, представленной на рис. 2.3.

ФКС КТСОП, используемые для обмена данными между абонентами ВС, можно, как и линии связи, разделить на *абонентские* и *служебные*.

В качестве ООД *абонентских* ФКС выступают подключаемые к ВС абонентские компьютеры или провайдерское оборудование с одной стороны, и концентраторы КТСОП – с другой. В качестве АПД служат модемы абонентских компьютеров или аналогичные им узлы и блоки коммуникационного оборудования сетевых провайдеров, с одной стороны, и блоки сопряжения концентраторов с АЛС (называемые *абонентскими комплектами*) – с другой. Роль передающей среды кабельных абонентских ФКС обычно выполняет электрический провод или кабель, а абонентских ФКС провайдеров – часто волоконно-оптический кабель. БУР в абонентских ФКС обычно отсутствуют. Ввиду существенно отличающейся от нуля нижней граничной частоты полосы пропускания абонентских ФКС КТСОП, кроме абонентских ФКС провайдеров, в качестве сигналов-носителей данных в них обычно используются не двух- и многоуровневые сигналы, нижняя граничная частота спектра которых равна нулю (см. подп. 2.1.2), а модулированные сигналы (п. 2.6). В абонентских ФКС провайдеров, ввиду отсутствия организационно-законодательных ограничений на их полосу пропускания, обычно применяются двух- и многоуровневые сигналы-носители [3].

Функции ООД *служебных* ФКС выполняют блоки сопряжения ОС и ПК с МЛС (см. рис. 2.6). В качестве передающей среды в них обычно используется волоконно-оптический кабель. При длине МЛС более нескольких километров, как правило, применяются промежуточные БУР. При использовании ВОК в качестве передающей среды служебных ФКС обычно применяются двухуровневые сигналы-носители данных.

Следует отметить, что из-за ограничения полосы пропускания абонентского ФКС КТСОП частотами от 300 до 3400 Гц (см. выше), предельно достижимая на практике скорость обмена данными между

абонентом ВС и провайдером по указанному ФКС обычно не превышает 56 000 бит/с [3] (подп. 2.6.5). Однако современный уровень развития сетевых технологий требует значительно более высоких скоростей, особенно при передаче данных от провайдера к абоненту. С другой стороны, указанное ограничение обусловлено организационно-законодательными, но не техническими факторами; в отсутствие этого ограничения ширина полосы пропускания абонентского ФКС составляет минимум несколько сотен килогерц, что позволяет достигнуть скоростей обмена минимум порядка нескольких сотен Кбит в секунду [7]. Поэтому в настоящее время достаточно широко распространены технологии обмена данными между абонентом КТСОП и провайдером, позволяющие устранить ограничения на скорость обмена, накладываемые полосой пропускания абонентского ФКС. Наиболее распространенными из них являются *технологии группы xDSL* [1, 3, 7].

Основой данного названия является аббревиатура английского словосочетания «Digital Subscriber Line», в переводе – «Цифровая абонентская линия». Индекс «x» различен для различных вариантов указанных технологий. В частности, известны такие их разновидности: *ADSL* (Asymmetric Digital Subscriber Line, в переводе – «Асимметричная цифровая абонентская линия»); *HDSL* (High bit-rate Digital Subscriber Line, в переводе – «Высокоскоростная цифровая абонентская линия»); *SDSL* (Single line Digital Subscriber Line, в дословном переводе – «Однолинейная цифровая абонентская линия») и некоторые другие [1, 7]. Общий принцип большинства технологий группы xDSL состоит в использовании АЛС КТСОП для связи как между абонентским телефонным аппаратом (ТА)/факсом и КТСОП, так и для высокоскоростного (порядка сотен Кбит – единиц Мбит в секунду) обмена данными между абонентским ПК и провайдером. Однако для решения двух указанных задач связи формируется два отдельных, независимых потока данных. Во избежание взаимного влияния двух указанных потоков, для их представления в АЛС применяется какой-либо из методов их *разделения* (*мультиплексирования*) (п. 2.8). Например, *технология ADSL*, достаточно широко применяемая для соединения физических лиц с ГВС Internet [7], использует *разделенные по частоте* сигналы-носители, спектры которых находятся в различных, не перекрывающихся между собой диапазонах частот. При этом для передачи по АЛС трафика между абонентским ТА/факсом

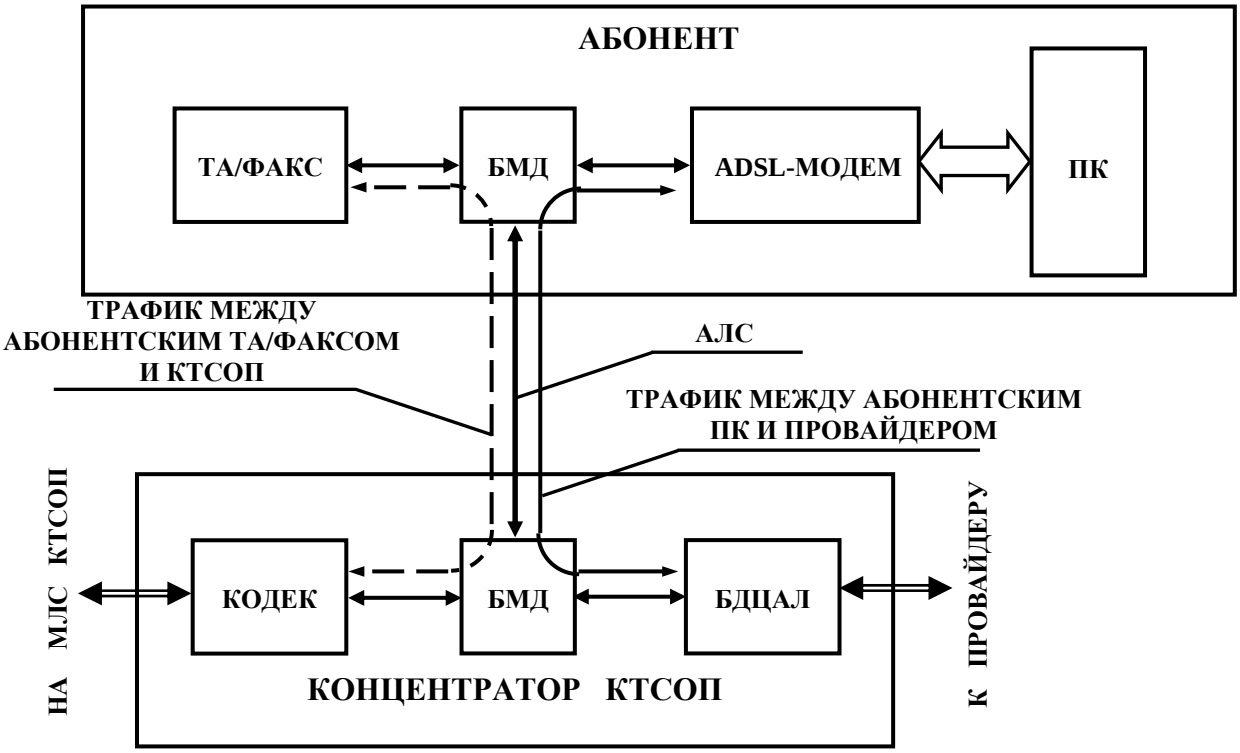
и КТСОП применяются сигналы, граничные частоты амплитудного спектра которых равны 300 и 3400 Гц соответственно. В свою очередь, для передачи по АЛС трафика между абонентским ПК и провайдером используются сигналы, спектр которых лежит в пределах от некоторой частоты f_{1DSL}, равной порядка нескольких десятков килогерц (т. е. существенно большей 3400 Гц) до частоты f_{2DSL}, приближенно равной физически достижимой верхней граничной частоте полосы пропускания конкретной АЛС. Указанная частота зависит в основном от типа кабеля АЛС и от расстояния между абонентом и концентратором КТСОП, и обычно составляет порядка нескольких сотен килогерц – единиц мегагерц [7]. В свою очередь, примерно 90 % частотного диапазона от f_{1DSL} до f_{2DSL} выделяется под сигналы-носители данных от провайдера к абоненту (т. е. под входящий трафик абонента, называемый по-английски *downstream*), а примерно 10 % указанного диапазона – под исходящий трафик абонента (*upstream*). При этом обеспечивается скорость передачи данных от провайдера к абоненту порядка сотен Кбит/с – единиц Мбит/с, а в обратном направлении – порядка десятков – сотен Кбит/с [7]. Большая ширина диапазона частот, выделяемого под входящий трафик абонента, чем под исходящий, обусловлена значительно меньшей интенсивностью последнего. В самом деле, на практике объем (в битах) запросов от абонента к провайдеру минимум на порядок меньше, чем объем данных, поступающих в ответ на эти запросы. Слово «Асимметричная» («Asymmetric») в названии ADSL-технологии обмена данными указывает именно на неодинаковую ширину диапазонов частот, выделяемых под входящий и исходящий трафики абонента.

Принцип ADSL-технологии поясняет рис. 2.7.

Носителем данных между абонентским ТА/факсом и КТСОП на участке от абонента до концентратора является аналоговый сигнал в частотном диапазоне от 300 до 3400 Гц (см. выше). Преобразование цифровых данных, поступающих с МЛС КТСОП, в указанный аналоговый сигнал, а также преобразование аналогового сигнала, поступающего с ТА/факса в последовательность цифровых отсчетов, передаваемую в МЛС, осуществляются посредством *кодека* (рис. 2.7) [7].

В качестве носителей данных между абонентским ПК и провайдером на участке от абонента до концентратора КТСОП обычно служат модулированные синусоидальные сигналы [1], формируемые посред-

ством БДЦАЛ и, соответственно, ADSL-модема (рис. 2.7). Эти же блоки осуществляют и обратное преобразование сигналов-носителей в цифровые данные. Методы модуляции, применяемые ADSL-технологией, вкратце будут рассмотрены в подп. 2.6.4.



ТА – телефонный аппарат;
БМД – блоки мультиплексирования/демультиплексирования;
ПК – персональный компьютер;
БДЦАЛ – блок доступа к цифровой абонентской линии

Рис. 2.7. Пояснение принципа реализации технологии ADSL

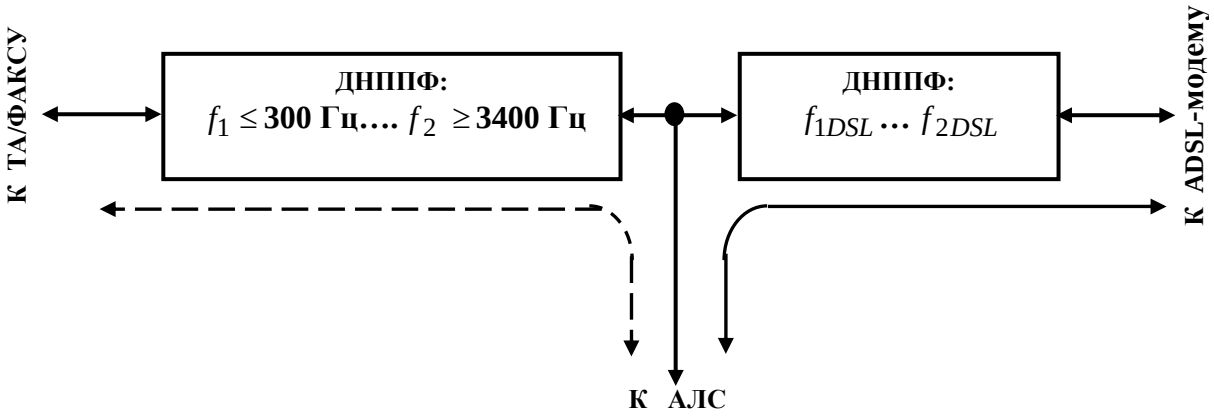
Упрощенная структурная схема БМД ADSL представлена на рис. 2.8. Мультиплексирование/демультиплексирование сигналов-носителей каждого из потоков данных как на стороне абонента, так и в концентраторе осуществляется посредством БМД (см. рис. 2.7), упрощенная структурная схема которого представлена на рис. 2.8. Он состоит из двух двунаправленных полосно-пропускающих фильтра (ДНППФ). Двунаправленными они названы потому, что осуществляют передачу сигнала (с фильтрацией) в обоих направлениях. Границы полосы пропускания каждого из ДНППФ указаны на рис. 2.8.

Интересно отметить, что xDSL-технологии, в отличие от связи с ВС через «традиционные» абонентские ФКС КТСОП, позволяют

в одно и то же время как пользоваться ТА/факсом, так и осуществлять обмен данными между абонентским ПК и провайдером.

Необходимо отметить, что, кроме кабельных АЛС и МЛС КТСОП, достаточно широко применяются *беспроводные* АЛС, известные под названием *беспроводных абонентских окончаний*, а также *беспроводные* МЛС. Краткий обзор принципов их реализации представлен в подп. 2.2.3.

До недавнего времени АЛС КТСОП широко применялись для подключения абонентов к ГВС, а также к сетям мегаполисов, в некоторых практических случаях – и ЛВС [3, 7]. В настоящее время для этого все шире используются выделенные (преимущественно волоконно-оптические) или беспроводные линии связи. Однако применение АЛС КТСОП для подключения к ГВС (в основном, с использованием xDSL-технологий) все еще достаточно распространено, особенно в относительно некрупных городах.



ДНППФ – двунаправленные полосно-пропускающие фильтры

Рис. 2.8. Упрощенная структурная схема БМД ADSL

МЛС КТСОП (как кабельные, так и беспроводные) относительно широко используются в качестве магистральных линий связи ГВС, а в ряде случаев – и ВС других классов.

2.2.2. Выделенные кабельные ФКС (ВКФКС)

Данный тип ФКС применяется для передачи данных или между абонентами ЛВС, или на отдельных участках ГВС. Реализация всей коммуникационной системы ГВС на основе выделенных ФКС является дорогостоящей и сложной с организационно-технической точки зрения, и поэтому достаточно редко применяется на практике, в основном, в крупных городах. Напомним, что использование только (или в основном) выделенных линий связи является основным отличительным признаком ЛВС от ГВС (см. подп. 1.1.2).

В соответствии с вышесказанным, ВКФКС можно разделить на два основных типа:

- кабельные ФКС ЛВС;
- ВКФКС, предназначенные для передачи информации на отдельных участках линий связи ГВС.

ФКС первого из перечисленных типов характеризуются относительно небольшой протяженностью (от десятков метров до единиц километров). В качестве их ООД обычно служат абонентские компьютеры или коммуникационное оборудование ЛВС, например, коммутаторы, а в качестве АПД – соответственно сетевые адаптеры абонентских компьютеров или аналогичные им блоки коммуникационного оборудования ЛВС. Организационно-законодательные ограничения на ширину полосы пропускания ФКС данного типа отсутствуют, и она определяется только параметрами и характеристиками их АПД, БУР и передающей среды. Функции последней в ФКС ЛВС обычно выполняет электрический или волоконно-оптический кабель (п. 2.3). В качестве носителей данных в них, как правило, применяются двух- или многоуровневые сигналы (см., например, рис. 2.2, *а*, *б*).

Кабельные ФКС, выделенные для передачи информации на отдельных участках линий связи ГВС, отличаются значительно большей протяженностью, чем ФКС ЛВС. Роль ООД в них играют узлы коммуникационной сети ВС, например, маршрутизаторы (гл. 4), функции АПД выполняют соответствующие блоки этих узлов, а в качестве передающей среды обычно применяется волоконно-оптический кабель. Ввиду значительной протяженности данный тип ФКС, как правило, снабжается одним или несколькими БУР (см. рис. 2.1). В качестве носителей данных в этих ФКС обычно используются двухуров-

невые сигналы. Для ФКС данного типа характерно уплотнение (мультиплексирование), как правило, временное или волновое (п. 2.8). В ряде практических случаев для обмена информацией между абонентами ГВС выделяется только часть каналов мультиплексированной линии связи [3].

2.2.3. Беспроводные ФКС общего пользования

Наиболее распространенными разновидностями ФКС данного типа являются [3, 7]:

- каналы системы мобильной телефонной связи (СМТС);
- ФКС беспроводных АЛС КТСОП, называемые также *беспроводными абонентскими окончаниями* (по-английски – *Wireless Local Loop, WLL*);
- ФКС беспроводных МЛС КТСОП (см. рис. 2.6).

Для передачи данных в ФКС всех вышеназванных разновидностей в основном используется электромагнитное излучение *микроволнового диапазона*, с частотами от сотен МГц до единиц ГГц. Применение данного диапазона позволяет выделять под ФКС полосы частот шириной от нескольких сотен кГц до десятков МГц. Использование более низкочастотных диапазонов пропорционально уменьшает ширину полосы частот, которая потенциально может быть выделена под ФКС и, соответственно, пропускную способность ФКС (см. выражения (2.7) и (2.8)). Однако уверенная связь в микроволновом диапазоне, как правило, возможна только в пределах прямого распространения радиоволн между приемником и передатчиком, что обусловливает определенную специфику организации ФКС рассматриваемого типа (см. далее).

Следует отметить, что для всех беспроводных ФКС общего пользования, как и для кабельных АЛС КТСОП, характерны *организационно-законодательные ограничения* на ширину полосы пропускания ФКС, очевидные при использовании беспроводной передающей среды, разделяемой множеством абонентов [3].

Основным назначением *первой* из вышеперечисленных разновидностей ФКС является обмен голосовыми сообщениями между абонентами СМТС. Однако в настоящее время эти ФКС достаточно широко применяются для передачи информации между абонентами ВС (преимущественно ГВС). Упрощенная структурная схема типового фрагмента СМТС представлена на рис. 2.9 [5, 7].

Данный фрагмент состоит:

- из ряда подсистем базовых станций (ПСБС), по-англ. – *Base Station Subsystems* (*BSS*);

- сетевой и коммутационной подсистемы (СКП), в англоязычной литературе – *Network and Switching Subsystem* (*NSS*);

- подсистемы управления сетью (ПСУС), по-англ. – *Network Management Subsystem* (*NMS*).

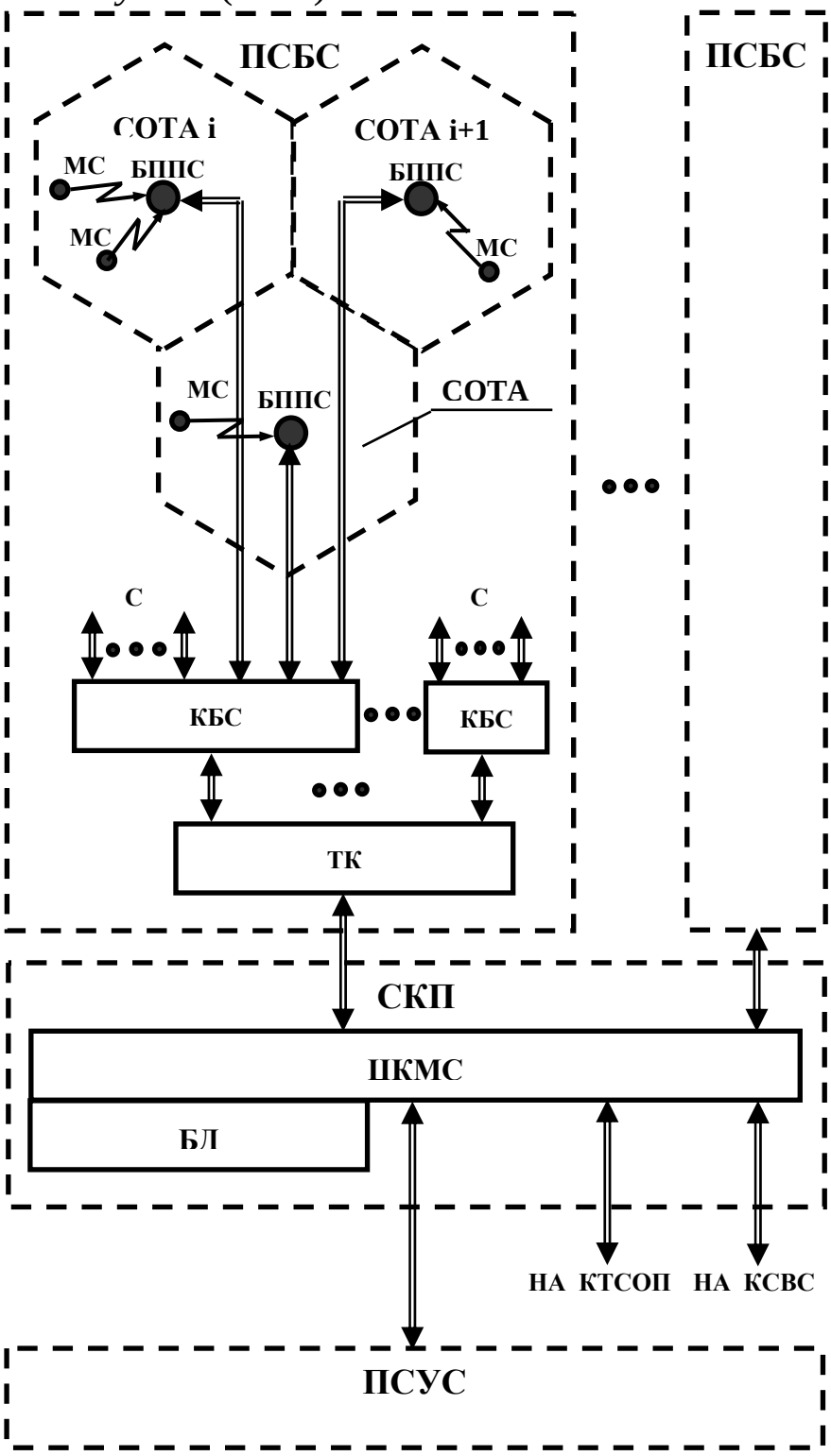


Рис. 2.9. Упрощенная структурная схема типового фрагмента СМТС (расшифровку аббревиатур см. в тексте)

Каждая из *ПСБС*, в свою очередь, включает в себя:

- набор базовых приемопередающих станций (БППС), *Base Transceiver Stations* (*BTS*);
- контроллеры базовых станций (КБС), *Base Station Controllers* (*BSC*);
- транскодер (ТК), *Transcoder* (*TCE*).

БППС формируют так называемые *соты* (*ячейки*) СМТС. Каждая из них представляет собой зону действия определенной БППС, т. е. область уверенной связи между находящимися в ее пределах *мобильными станциями*, МС (сотовыми телефонами и т. п.) и соответствующей БППС. Показано, что при круговой диаграмме направленности приемопередающей антенны БППС (что всегда имеет место на практике) оптимальной формой такой области является правильный шестиугольник [1, 5, 7] (см. рис. 2.9). При такой форме ячейки, во-первых, БППС обеспечивает доступ ко всем ее участкам, а, во-вторых, – практически отсутствуют перекрытие и пропуск ячеек СМТС. Из-за сходства формы ячейки с пчелиной сотой она также называется сотой, а мобильная телефонная связь – сотовой. Естественно, на практике формы ячеек несколько отличаются от правильного шестиугольника и зависят от рельефа местности, характера застройки и т. п. БППС, насколько позволяет местность, располагаются в геометрических центрах сот. Радиус соты, в зависимости от конкретных местных условий, может составлять от нескольких сотен метров до 30 – 35 км [5, 7].

Основной функцией БППС является радиосвязь с МС, находящимися в пределах обсуживаемой ею соты. Связь между МС, территориально относящимися к одной и той же соте, осуществляется через ее БППС. Обмен данными между МС, расположенными в различных сотах, реализуется по тракту, формируемому БППС этих сот, а также КБС, к которому они подключены. Последний представляет собой коммутационно-управляющее устройство на основе достаточно мощного и производительного компьютера, выполняющее функции управления потоками данных между сотами, а также управления и контроля БППС. Как правило, один КБС обслуживает несколько десятков БППС и, соответственно, сот [3, 7].

ТК представляет собой, по существу, блок интерфейса между ПСБС и СКП (см. далее), одной из основных функций которой является сопряжение СМТС с КТСОП и коммутационными системами ВС (КСВС). ТК осуществляет взаимное преобразование данных,

представляемых в соответствии со стандартами СМТС, и данных, представляемых в соответствии со стандартами КТСОП и КСВС.

СКП осуществляет коммутацию как между ПСБС, входящими в состав СМТС, так и между СМТС, с одной стороны, и КТСОП и КСВС – с другой. Данные функции реализуются посредством входящего в состав СКП *центра коммутации мобильных станций* (ЦКМС), по-англ. – *Mobile Services Switching Center* (MSC). Также СКП содержит базы данных (БД), необходимые для управления мобильностью абонентов и обеспечения безопасности сети [3, 7]. Основными элементами указанных БД являются:

- домашний регистр, содержащий сведения обо всех абонентах, зарегистрированных в соответствующей сети МТС (т. е. у соответствующего оператора МТС);
- гостевой регистр, который содержит аналогичные сведения об абонентах других сетей (т. е. других операторов МТС), но пользующихся услугами данной сети;
- регистр идентификации оборудования, содержащий сведения об оборудовании МС, в частности, данные для подтверждения подлинности международных идентификационных номеров МС, содержащихся в SIM-картах.

ПСУС включает в себя [5]:

- центр эксплуатации и технического обслуживания;
- центр управления сетью.

Связь между МС и БППС осуществляется по радиоканалам. Другие линии связи СМТС, как правило, являются кабельными [1] и, в целом, аналогичны МЛС КТСОП (см. рис. 2.6 и пояснения к нему) как по типам применяемой передающей среды, так и по способам представления данных.

Для радиосвязи между МС и БППС различные стандарты МТС используют диапазоны частот, находящиеся в пределах от порядка 800 – 900 МГц до единиц ГГц. Например, согласно базовому стандарту GSM, для передачи данных от МС к БППС (*Uplink*) выделен частотный диапазон от 890,2 до 914,8 МГц, а для передачи от БППС к МС (*Downlink*) – диапазон частот от 935,2 до 959,8 МГц.

Разделение множеством МС одного и того же участка передающей среды в пределах соты осуществляется сочетанием методов частотного и временного мультиплексирования (в более ранних стандартах мобильной связи) или в современных стандартах – кодового, частотного и временного (п. 2.8).

При каждом из двух вышеописанных подходов граничные частоты полосы пропускания физических каналов связи между МС и БППС существенно отличны от нуля, ширина полосы пропускания минимум на порядок меньше значений указанных частот. Поэтому в качестве носителей данных в этих ФКС применяются модулированные синусоидальные сигналы, для которых, в отличие от двух- и многоуровневых сигналов (см. рис. 2.2), возможно обеспечить совпадение границ спектра с границами полосы пропускания физических каналов связи между МС и БППС.

Как и КТСОП, СМТС применяется не только для обмена речевыми сообщениями между абонентами, но и в качестве элемента системы связи ГВС, ВС мегаполисов, в некоторых практических случаях – и ЛВС [3]. При этом абоненты СМТС выступают также в качестве абонентов ВС, используя МС в качестве ООД и приемопередатчики МС – в качестве АПД.

Другой распространенной разновидностью беспроводных ФКС общего пользования являются ФКС *беспроводных абонентских окончаний* (WLL). WLL, по существу, представляют собой беспроводные АЛС КТСОП (см. рис. 2.6), предназначенные для подключения абонентов КТСОП к ее коммутационному оборудованию в тех случаях, когда прокладка проводных АЛС затруднительна или невозможна. Это имеет место, например, в труднодоступной местности или при высокой стоимости лицензии на прокладку проводных АЛС.

Известны следующие основные принципы реализации WLL [3, 5]:

- стационарный (фиксированный) радиодоступ;
- стационарная сотовая связь;
- микроволновая линия типа «точка – много точек».

Системы *стационарного радиодоступа*, называемые также *зоновыми системами*, обычно служат для подключения к КТСОП абонентов, распределенных в пределах прямого распространения радиоволн (в радиусе порядка единиц – десятков километров) вокруг ОС КТСОП. Данная WLL-технология практически является непосредственным беспроводным аналогом технологий обмена данными по проводным АЛС КТСОП. Она базируется на прямом соединении абонентов КТСОП с ее коммутационным оборудованием (концентраторами и ОС) по выделяемым для этой цели радиоканалам, граничные частоты которых, методы представления данных и уплотнения каналов связи определяются конкретным стандартом WLL [3]. Стацио-

нарный радиодоступ применяется в основном для замены проводных АЛС при подключении к КТСОП абонентов пригородных и ближних сельских районов.

Принцип *стационарной сотовой связи* базируется на применении технологий мобильной телефонной связи (см. выше) для подключения абонентов к КТСОП. В основном структура системы стационарной сотовой связи и применяемые в ней методы представления и передачи данных аналогичны используемым в СМТС (см. выше). Их основным отличием является то, что первая из названных систем обычно не предусматривает перемещения абонента в пространстве, т. е. предполагает, что он является *стационарным*, откуда и происходит название указанной системы. Системы стационарной сотовой связи применимы как в городских, так и пригородных и сельских районах. Обычно они используются для обслуживания районов с относительно большой площадью и неравномерным распределением абонентов. Однако их стоимость достаточно высока из-за сравнительно большого количества сот, требуемого для практической реализации данных систем [3].

Микроволновые линии типа «точка – много точек» применяются в основном для подключения к КТСОП абонентов малонаселенной сельской местности [3]. Обычно они представляют собой *радиорелейные линии связи* (РРЛС) с многолинейными ответвлениями, в качестве которых, в свою очередь, выступают АЛС. Пример структурной схемы фрагмента микроволновой линии данного типа представлен на рис. 2.10. Она образована последовательностью ретрансляционных радиостанций (РРС), работающих в *микроволновом* частотном диапазоне (порядка единиц – десятков ГГц), и располагаемых на расстоянии прямого распространения радиоволн друг от друга. Благодаря тому, что антенны РРС размещаются на относительно большой высоте (порядка нескольких десятков метров), указанное расстояние обычно составляет несколько десятков километров. Антенны РРС являются направленными (в большинстве случаев – параболическими), пространственно сориентированными на соседние РРС, что позволяет осуществлять передачу сигналов между ними с минимальными потерями. В целом, последовательность РРС формирует беспроводный тракт передачи сигналов на достаточно дальние расстояния (до нескольких сотен километров).

Каждая РРС может выполнять не только собственно функции ретрансляции, но и подключения абонентов к РРЛС (рис. 2.10). Посред-

ством контроллера РРЛС (см. там же) осуществляется сопряжение РРЛС с КТСОП и, следовательно, подключение к КТСОП або-нентов РРЛС.

При малой плотности абонентов и значительных расстояниях между пунктами их сосредоточения применение РРЛС значительно выгоднее, с экономической точки зрения, чем прокладка кабельных линий связи [3].

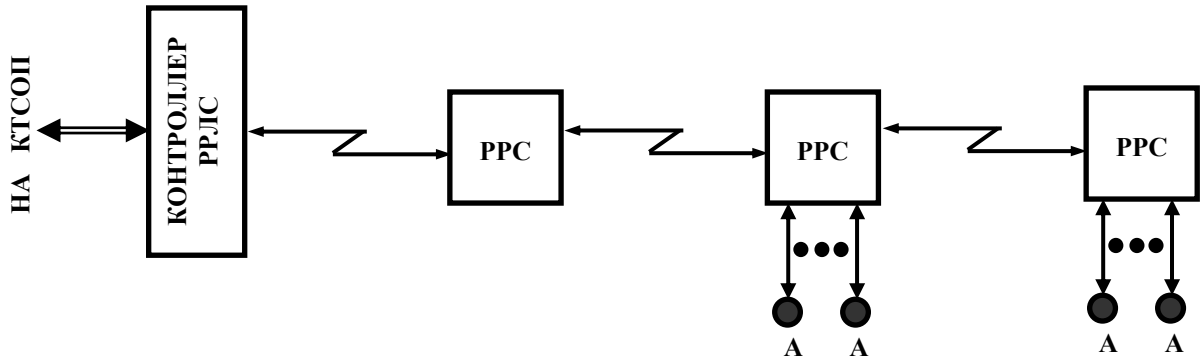


Рис. 2.10. Пример структурной схемы фрагмента микроволновой линии типа «одна точка – много точек» (расшифровку аббревиатур см. в тексте)

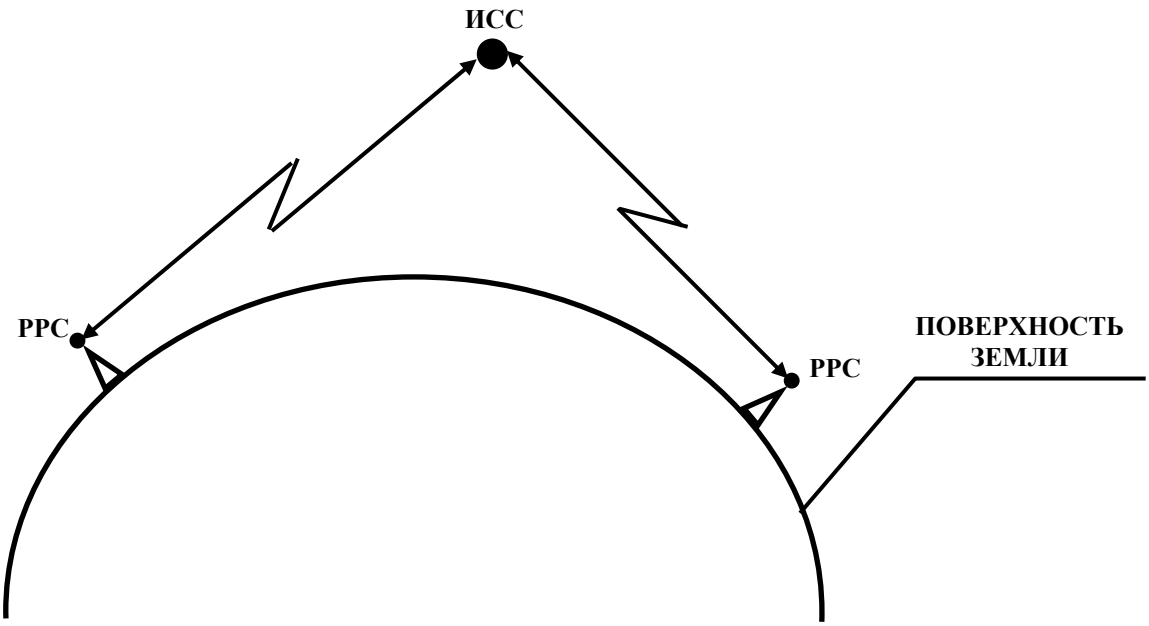
Частотные диапазоны, выделяемые под WLL, в зависимости от конкретного типа WLL-системы находится в областях от нескольких сотен МГц до десятков ГГц при ширине частотного диапазона от со-тен кГц до десятков МГц [3]. Поэтому, как и в каналах связи МС с БППС (см. выше), в качестве носителей данных в WLL применяют-ся модулированные синусоидальные сигналы, с использованием раз-личных технологий уплотнения ФКС (п. 2.8): частотного (например, в WLL-системе Telecell), временного (WLL-система Proximity), орто-гонального частотного (технология WiMAX), частотного в сочетании с временным (WLL-системы S-WLL и DRA1900), а также кодового (WLL-системы AirLoop и Airspan 60) [3, 5].

Аналогично проводным АЛС, WLL достаточно широко применя-ются не только для обмена речевыми сообщениями между абонента-ми КТСОП, но и для подключения абонентских компьютеров к ГВС (в ряде случаев – ВС мегаполисов и ЛВС). При этом в качестве або-нентских ООД и АПД служат компьютеры и WLL-модемы соответ-ственно.

Беспроводные МЛС КТСОП, как и беспроводные АЛС, применя-ются в тех случаях, когда прокладка кабельных линий невозможна,

затруднительна или экономически невыгодна. Большинство современных МЛС данного типа представляют собой *РРЛС* (см. выше) или *спутниковые* линии связи, в ряде случаев – их сочетания. Принцип реализации служебных РРЛС КТСОП в целом аналогичен принципу реализации микроволновых линий типа «одна точка – много точек» (см. рис. 2.10 и пояснения к нему), за исключением того, что РРС служебных РРЛС КТСОП часто не имеют многолинейных ответвлений [3].

Спутниковые МЛС КТСОП, как правило, также представляют собой разновидность РРЛС, функции одной (значительно реже – нескольких) из РРС которых выполняют искусственные спутники Земли, часто – с *геостационарной* орбитой, обеспечивающей неизменное во времени положение спутника относительно земной поверхности [3]. Их применение в качестве РРС позволяет весьма существенно (до нескольких тысяч километров) увеличить расстояние между соседними *наземными* РРС, что иллюстрирует рис. 2.11.



ИСС – искусственный спутник связи

Рис. 2.11. Структурная схема фрагмента РРЛС с искусственным спутником связи в качестве РРС

Это обусловлено несопоставимо большей высотой расположения спутника над поверхностью Земли (от сотен до десятков тысяч километров), чем наземных РРС (десятки метров) и, как следствие, значи-

тельно большей площадью области уверенной связи со спутником, чем с наземной РРС. Благодаря данному свойству спутниковых каналов связи, они обычно применяются для формирования участков РРЛС протяженностью порядка сотен – тысяч километров, на которых строительство наземных РРС или прокладка кабеля затруднительна или невозможна, например, акватории морей и океанов, труднодоступные или непроходимые участки суши и т. п.

В качестве сигналов-носителей данных в беспроводных магистральных ФКС применяются модулированные синусоидальные сигналы микроволнового диапазона частот. Ширина выделяемого для связи частотного диапазона обычно составляет порядка нескольких десятков – сотен МГц [3].

Как и другие линии связи КТСОП, беспроводные МЛС КТСОП используются не только для передачи потоков речевых сообщений (естественно, представленных в цифровой форме), но и в качестве магистральных линий связи коммуникационной системы ВС.

2.2.4. Выделенные беспроводные ФКС (ВБФКС)

К данной категории относятся следующие основные типы ФКС:

- беспроводные ФКС ЛВС;
- беспроводные магистральные ФКС ГВС.

Беспроводные ФКС ЛВС, как следует из их названия, служат для обмена данными между абонентами ЛВС и/или ее коммуникационным оборудованием. Как и другие типы ФКС ЛВС, они характеризуются малой протяженностью (порядка десятков – сотен метров). Для передачи данных используется электромагнитное излучение микроволнового или инфракрасного диапазона.

Одними из наиболее распространенных стандартов обмена данными по беспроводным ФКС ЛВС являются стандарты группы IEEE 802.11 (более известные под названием стандартов группы Wi-Fi) [3, 5]. Они оговаривают протоколы обмена данными по беспроводным ФКС ЛВС на физическом и канальном уровнях логической модели ВС. Физический уровень стандартов группы IEEE 802.11 предполагает использование для связи одного из двух частотных диапазонов:

- микроволнового излучения с частотами, лежащими в области значений 2,4, 3,6 или 5 ГГц, при ширине выделяемого под связь частотного диапазона, равной нескольким десяткам МГц;

- инфракрасного излучения с длиной волны от 850 до 950 нм.

При использовании микроволнового частотного диапазона в качестве носителей данных используются модулированные синусоидальные сигналы, а инфракрасного диапазона – импульсные сигналы.

Типовая структурная схема Wi-Fi-сегмента ЛВС представлена на рис. 2.12. Сегментообразующим блоком является *точка доступа Wi-Fi*, функционально аналогичная коммутатору ЛВС с кабельной средой передачи (п. 3.7). В ФКС данного сегмента в качестве ООД служат абонентские компьютеры (АК) и, соответственно, точка доступа, а в качестве АПД – Wi-Fi-модемы (МД) абонентских компьютеров и точки доступа.

Максимальная протяженность ФКС при использовании для связи микроволнового диапазона частот обычно составляет порядка нескольких десятков метров, а инфракрасного диапазона – порядка нескольких метров.

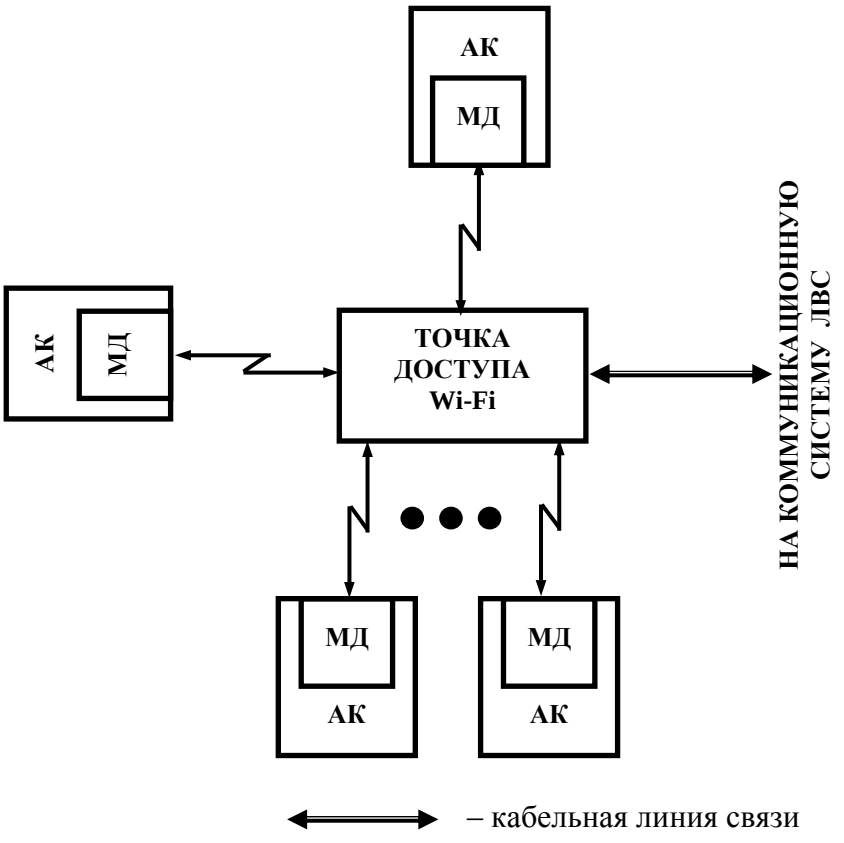


Рис. 2.12. Типовая структурная схема Wi-Fi-сегмента ЛВС (расшифровку аббревиатур см. в тексте)

Беспроводные магистральные ФКС ГВС обычно служат для обмена потоками данных между узлами (преимущественно крупными)

коммуникационной системы ГВС при сложности, невозможности или экономической нецелесообразности прокладки кабельных магистралей между этими узлами. Данный тип ФКС, как и беспроводные МЛС КТСОП, обычно реализуется или на основе РРЛС (см. рис. 2.10), или на базе спутниковых систем связи (см. рис. 2.11) [3]. Необходимо отметить, что искусственные спутники связи при этом могут выполнять функции не только РРС, но и маршрутизаторов ГВС (гл. 4).

В качестве сигналов-носителей данных в беспроводных магистральных ФКС, как и в беспроводных ФКС других типов, применяются модулированные синусоидальные сигналы микроволнового диапазона частот [3, 5].

2.3. Передающая среда ФКС

Основными типами передающей среды современных ВС являются [3, 7]:

- проводные линии связи;
- электрические кабели;
- волоконно-оптические кабели (ВОК);
- беспроводная передающая среда.

2.3.1. Проводные линии связи как передающая среда ФКС ВС

Данный тип передающей среды представляет собой пару изолированных проводов, без каких-либо экранирующих элементов, наличие которых, в свою очередь, является отличительной особенностью электрических кабелей (подп. 2.3.2). До недавнего времени проводные линии связи были основным типом передающей среды КТСОП и телеграфных линий. Ограниченно применяются они и в настоящее время, в основном – для связи абонентов КТСОП, в том числе абонентов ВС, подключаемых к ней через КТСОП, с концентратором (оконечной телефонной станцией) КТСОП.

Данный тип передающей среды характеризуется наименьшей стоимостью из всех перечисленных, однако, с другой стороны – наименьшей помехозащищенностью. Полоса пропускания (затухание) проводных линий в целом позволяют передавать по ним сигналы в частотном диапазоне от 0 до порядка 1 – 1,5 МГц [7]. Необходимо однако отметить, что для каналов связи абонентов КТСОП с оконечной телефонной станцией (концентратором) по проводной линии существует *организационно-законодательное ограничение* на граничные частоты полосы пропускания (см. подп. 2.2.1). Данное ограничение, однако, устраняется, например, при использовании xDSL-технологий (см. там же).

2.3.2. Электрические кабели как передающая среда ФКС ВС

В качестве передающей среды данного типа могут служить два вида кабелей:

- кабели на основе витых пар проводов;

- коаксиальные кабели.

Коаксиальные кабели до недавнего времени достаточно широко применялись в качестве передающей среды ФКС ВС (в, частности, ЛВС). Однако в настоящее время в ФКС протяженностью порядка десятков – сотен метров коаксиальные кабели практически полностью вытеснены витыми парами, а большей протяженности – ВОК [3].

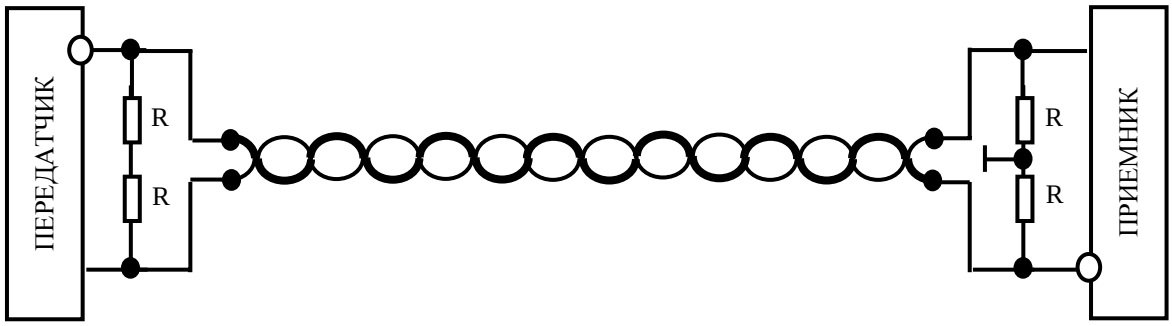
Кабели на основе витых пар. Существуют две основные разновидности кабелей данного типа: на базе *неэкранированных* и *экранированных* витых пар.

Неэкранированная витая пара, по-английски – *Unshielded Twisted Pair (UTP)* представляет собой пару проводов, скрученных между собой с определенным шагом, зависящим от категории UTP-кабеля (см. далее). Упрощенный вид UTP представлен на рис. 2.13, *а*. UTP-кабели обычно содержат несколько витых пар, помещенных внутрь общей защитной оболочки из непроводящего материала. В частности, достаточно широкое распространение получили *4-парные* UTP-кабели (т. е. содержащие 4 витые пары в общей оболочке).

Возможны два варианта передачи сигнала по UTP. По первому из них один из проводов пары является общим, а другой – сигнальным. За счет скручивания общего провода с сигнальным достигается эффект частичного экранирования последнего. Второй вариант предполагает *дифференциальную передачу сигнала* по UTP, в соответствии со схемой, представленной на рис. 2.13, *б*. Благодаря тому, что оба провода пары имеют одинаковую длину и проложены рядом, помехи наводятся на них практически одинаково, и в результате не влияют на разностный (дифференциальный) сигнал между проводами. Дифференциальная передача сигнала по UTP несколько более распространена на практике [3, 7].

Согласно международным стандартам, UTP-кабели подразделяются на 7 категорий, различающихся между собой конструктивными параметрами (в частности, шагом скрутки) и, как следствие, электрическими характеристиками.

а



$2R = R_B$, где R_B – волновое сопротивление кабеля

б

Рис. 2.13. Упрощенный вид неэкранированной витой пары (*а*) и схема ее подключения при дифференциальной передаче сигнала (*б*)

Основные параметры UTP-кабелей распространенных категорий будут представлены в приведенной далее сводной таблице параметров кабелей (табл. 2.1).

По сравнению с кабелями на основе экранированной витой пары, UTP-кабели характеризуются большим удобством прокладки и монтажа кабельной системы и меньшей стоимостью, однако, с другой стороны – меньшей помехозащищенностью [3, 7].

Экранированная витая пара, по-англ. – *Shielded Twisted Pair (STP)* отличается от неэкранированной тем, что ее проводники помещаются в проводящий заземляемый (зануляемый) электромагнитный экран. Если кабель на основе экранированной витой пары (STP-кабель) содержит несколько пар проводов в общей оболочке, в отдельный экран помещается каждая из них. Кроме данного вари-анта конструкции экранированного кабеля, называемого собственно STP-кабелем, известны также:

- *FTP-кабель* (от англ. словосочетания *Foiled Twisted Pair*), состоящий из нескольких неэкранированных витых пар, помещенных в общий экран, обычно изготавливаемый из металлической фольги (называемой по-англ. *foil*); данный тип кабеля называют также *S/UTP-кабелем*;

- *S/FTP-кабель*, состоящий из нескольких неэкранированных витых пар, помещенных в общий двойной экран, внутренний экраниру-

ющий слой которого обычно изготавливается из фольги, а внешний – из медной оплетки;

- *S/STP-кабель* (STP-кабель с двойным экранированием), который состоит из нескольких экранированных витых пар, помещаемых, в свою очередь, в общий экран.

Таблица 2.1

Основные параметры распространенных типов кабелей на основе витой пары

Тип передающей среды	Затухание (см. выражение (2.10)):			NEXT (см. выражение (2.11)):	
	длина отрезка, м	f_T, МГц	типовое значение, дБ	f_T, МГц	типовое значение, дБ
4-парный UTP категории 3 (R_B = 100 Ом)	100	0,064	- 0,9	0,15	- 53
		1	- 2,6	1	- 41
		10	- 9,8	10	- 26
		16	- 13,1	16	- 23
4-парный UTP категории 5 (R_B = 100 Ом)		1	- 2,0	1	- 62
		10	- 6,5	10	- 47
		31,25	- 11,7	31,25	- 39
		100	- 22,0	100	- 32
4-парный UTP категории 6 (R_B = 100 Ом)		1	- 2,3	1	- 62
		10	- 6,9	10	- 47
		100	- 23,0	100	- 38
		300	- 46,8	300	- 31
4-парный STP категории 6 (R_B = 100 Ом)	100	1	- 2,8	1	- 85
		10	- 8,6	10	- 85
		100	- 28,0	100	- 80
		300	- 50,5	300	- 75
4-парный S/FTP категории 6 (R_B = 100 Ом)		1	- 2,1	1	- 80
		10	- 6,0	10	- 80
		100	- 19,0	100	- 70
		300	- 33,0	300	- 70

Все перечисленные типы STP-кабеля, естественно, покрываются внешней защитной оболочкой из непроводящего материала.

При передаче сигналов посредством STP, как и при использовании UTP, на практике применяется и дифференциальная передача (см. рис. 2.13, *б*), и использование одного из проводников пары в качестве сигнального, а другого – в качестве общего.

Экранирование повышает помехозащищенность передаваемых данных (в том числе подавление взаимных наводок витых пар, объединенных в кабель), однако увеличивает стоимость кабеля и сложность его прокладки.

В заключение необходимо отметить, что экран STP-кабеля или экранирующий провод UTP-кабеля необходимо подключать к общей шине *только одной из соединяемых кабелем единиц АПД*, т. е. «зануллять» экран только на одной стороне ФКС. При подключении экрана к общим шинам обеих единиц АПД из-за неизбежной разности потенциалов их общих шин в экранирующем проводе возникают так называемые «блуждающие токи», которые создают серьезные наводки при обмене сигналами между АПД; известны даже случаи вывода АПД из строя этими наводками [3, 7].

Типовые параметры распространенных разновидностей кабелей на основе витых пар представлены в табл. 2.1 [3].

2.3.3. ВОК как передающая среда ФКС ВС

ВОК состоит из набора оптических волокон, основу каждого из которых составляет гибкий прозрачный сердечник (световод) диаметром порядка от единиц до десятков микрон, обычно изготавливаемый из специальных сортов стекла. Носителями информации в ВОК являются передаваемые по световодам импульсы электромагнитного излучения в так называемой ближней инфракрасной части спектра, с длиной волны, находящейся в области значений 850, 1300 или 1550 нм [3, 7]. Наличие импульса соответствует логической единице, а его отсутствие – нулю. Генерация импульсов на передающей стороне осуществляется электрически управляемым источником излучения – светодиодом или миниатюрным полупроводниковым лазером. На приемной стороне указанные импульсы посредством фотодиода преобразуются в электрические сигналы, поступающие на приемник АПД.

Сердечник покрывается слоем прозрачного материала (обычно также стекла), оптически менее плотного, чем материал сердечника, т. е. с меньшим показателем преломления. Данное покрытие позволяет значительно снизить потери излучения при его прохождении по световоду. Это достигается за счет эффекта *полного внутреннего отражения* [1]. Его сущность состоит в следующем. Если волна излучения из оптически более плотной среды (например, сердечника) падает

под некоторым углом α на границу ее раздела с оптически менее плотной средой (например, покрытием), то при условии, что:

$$\sin\alpha \rhd n_1/n_2 ,$$

где n_1 и n_2 – абсолютные показатели преломления оптически менее плотной и более плотной среды соответственно, преломленная волна отсутствует, и энергия падающей волны полностью возвращается в среду с более высоким показателем преломления.

В свою очередь, сердечник и его покрытие помещаются в непрозрачную защитную оболочку (обычно пластиковую).

Поперечное сечение оптического волокна представлено на рис. 2.14.

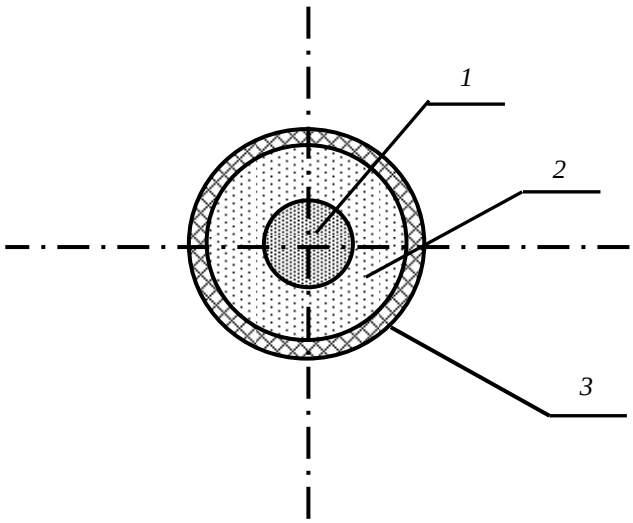


Рис. 2.14. Поперечное сечение оптического волокна (не в масштабе):
1 – сердечник (световод);
2 – покрытие сердечника;
3 – защитная оболочка

ВОК обычно состоит из нескольких (часто – нескольких десятков) оптических волокон, помещенных в общий защитный футляр. Следует заметить, что оптические волокна не являются источниками электромагнитных наводок, в том числе взаимных; поэтому не существует принципиальных ограничений на число волокон, объединяемых в ВОК.

По соотношению между диаметром сердечника и длиной волны излучения, используемого для передачи информации, различают *многомодовое* и *одномодовое* оптическое волокно (по-англ. – соответственно *Multi Mode Fiber*, *MMF*, и *Single Mode Fiber*, *SMF*) [3].

Диаметр сердечника *многомодового* волокна намного больше длины указанной волны (см. выше) и составляет порядка нескольких десятков микрон. Поэтому при единственном источнике излучения существует множество траекторий распространения его волны по световоду. Каждая из таких траекторий называется *модой*, откуда и происходит название «многомодовое волокно». Известны две разновидности многомодового оптического волокна – со ступенчатым и с градиентным профилями показателя преломления сердечника. У первой из них указанный показатель постоянен по всему сечению сердечника, а у второй – плавно уменьшается от центра к границе «сердечник-покрытие». На рис. 2.15 представлены примеры траекторий распространения волны излучения в многомодовом волокне со ступенчатым профилем показателя преломления сердечника.

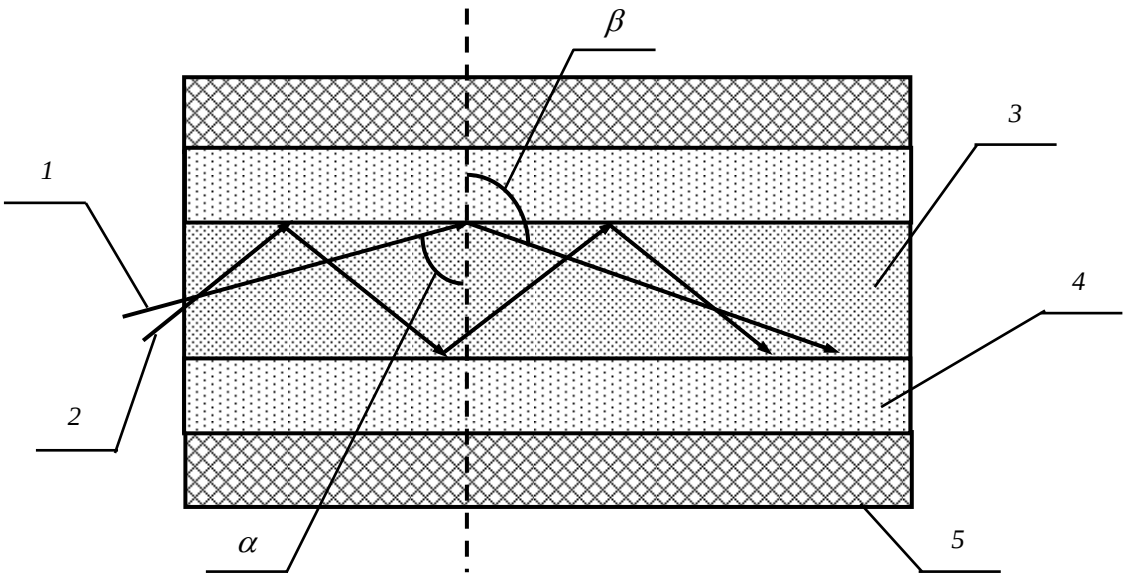


Рис. 2.15. Траектории распространения волны излучения в многомодовом оптическом волокне со ступенчатым профилем показателя преломления сердечника (не в масштабе):
1, 2 – моды (*1* – более высокого порядка, *2* – более низкого);
3 – сердечник (световод); *4* – покрытие сердечника;
5 – защитная оболочка; α, β – соответственно угол падения и угол преломления волны излучения

Рис. 2.15 также иллюстрирует принцип полного внутреннего отражения (см. выше).

Из рис. 2.15 видно, что каждая из мод характеризуется различным временем распространения по сердечнику (световоду). Это приводит к интерференции волн, соответствующих каждой из мод, на приемной стороне световода и, как следствие, к расширению во времени выход-

ного импульса фотоприемника по сравнению с переданным импульсом. Действительно, начало выходного импульса определяется моментом детектирования моды наименьшего порядка, а конец – моды наивысшего порядка (см. рис. 2.15). Данный эффект известен под названием *межмодовой дисперсии* [3]. На рис. 2.16 представлен типовой пример временных диаграмм входного и выходного импульсов световода при наличии межмодовой дисперсии. Нетрудно заметить, что ее результатом является снижение скорости передачи данных, которая, очевидно, ограничивается величиной, обратной ширине выходного
(а не входного) импульса.

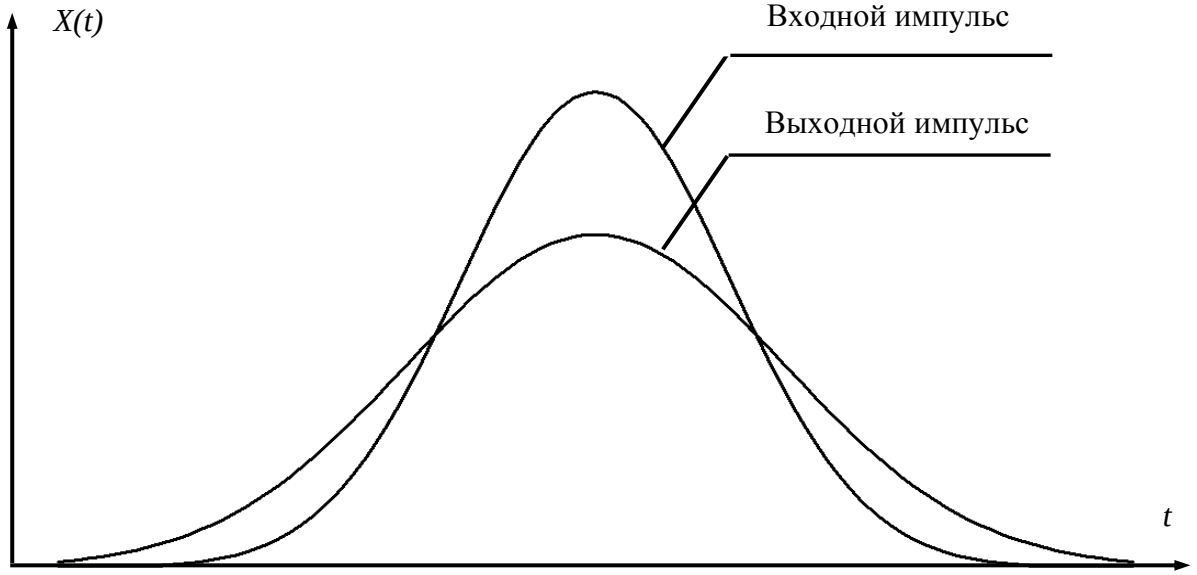


Рис. 2.16. Примеры временных диаграмм входного и выходного импульсов световода при наличии межмодовой дисперсии

Эффект межмодовой дисперсии менее выражен у волокна с градиентным профилем показателя преломления сердечника, чем у волокна со ступенчатым профилем. Поэтому оптическое волокно с градиентным профилем позволяет обеспечить более высокую скорость передачи данных. Однако оно отличается более сложной технологией изготовления и, следовательно, более высокой стоимостью, чем волокно со ступенчатым профилем.

Одномодовое оптическое волокно характеризуется диаметром сердечника, сопоставимым с длиной волны и равным порядка нескольких микрон. Благодаря этому сердечник представляет собой волновод, в котором практически отсутствует отражение волны от

границы «сердечник-покрытие», а траектория ее распространения является единственной и практически прямолинейной (рис. 2.17) [3].

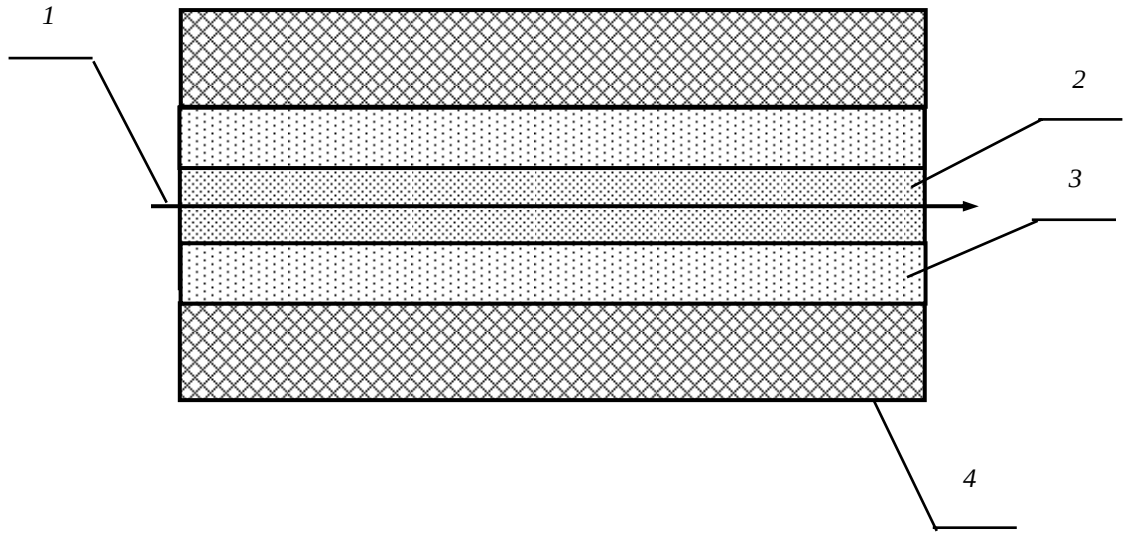


Рис. 2.17. Траектория распространения волны излучения в одномодовом оптическом волокне:
1 – мода; *2* – сердечник; *3* – покрытие сердечника; *4* – защитная оболочка

В одномодовом волокне, естественно, отсутствует эффект межмодовой дисперсии и обусловленное им расширение выходного импульса световода. Поэтому одномодовое волокно обеспечивает минимум в несколько раз более высокую скорость передачи данных, чем многомодовое. Кроме того, одномодовое волокно характеризуется существенно меньшими потерями сигнала (излучения) на единицу длины, чем многомодовое (см. далее). Вследствие этого одномодовое волокно обеспечивает передачу сигнала (без ретранслятора) на расстояние порядка нескольких десятков километров, в то время как многомодовое – примерно на порядок меньше [3].

Основным недостатком одномодового волокна является сложность производства и, как следствие, высокая стоимость.

В целом, из всех типов передающей среды ВКЛС, ВОК обеспечивает наивысшую скорость передачи данных, наименьшее затухание сигнала на единицу длины и наилучшую помехоустойчивость, обусловленную нечувствительностью информативного сигнала ВОК к электромагнитным помехам. Стоимость ВОК, в общем, сопоставима со стоимостью электрических кабелей ВКЛС. Однако стоимость прокладки и монтажа ВОК существенно выше, чем у них [3].

Типовые параметры ВОК представлены в табл. 2.2 [3].

Таблица 2.2

Основные параметры ВОК

Тип передающей среды	Затухание (см. выражение (2.10)):			NEXT (см. выражение (2.11)):	
	длина отрезка, м	длина волны, нм	типовое значение, дБ	f_T, МГц	типовое значение, дБ
Одномодовое оптическое волокно	1000	1310	- 1,0	Для оптоволокна значения данного параметра пренебрежимо малы	
		1550			
Многомодовое оптическое волокно		850	- 3,5		
		1300	- 1,0		

2.3.4. Беспроводная передающая среда ФКС ВС

Как следует из названия данного типа среды, она используется в беспроводных ФКС. В ее качестве обычно выступают воздух или безвоздушное пространство (в системах спутниковой связи).

Базовыми отличиями беспроводной передающей среды от электрического и оптоволоконного кабелей являются следующие [3]:

- беспроводная среда по своей естественной природе является *разделяемой* между теоретически неограниченным количеством ФКС, потенциально имеющих возможность использовать для связи один и тот же участок воздушного или безвоздушного пространства (эфира);
- беспроводная передающая среда является *ненаправленной*, т. е., вообще говоря, позволяющей распространяться сигналу-носителю (электромагнитным волнам) во всех направлениях.

Разделяемый характер беспроводной передающей среды обусловливает необходимость принятия специальных мер для устранения взаимного влияния ФКС, использующих один и тот же участок среды для обмена данными. До недавнего времени основным методом решения данной задачи являлось разделение ФКС *по частоте*, т. е. применение различных частотных диапазонов сигналов-носителей данных в различных ФКС. Простейшим примером такого разделения является использование радиостанциями, вещающими в одном и том же регионе, несущих сигналов с различными частотами. В настоящее время обычно применяется сочетание частотного разделения с временным и с кодовым (п. 2.8).

Ненаправленный характер беспроводной среды во многих случаях играет положительную роль, например, при формировании сот в СМТС (см. рис. 2.9). В ряде случаев, однако, необходимо формирование направленных беспроводных линий связи, типовым примером которых являются РРЛС (см. рис. 2.10). Обычно оно осуществляется за счет применения антенн, обеспечивающих прием и передачу сигналов только в пределах узконаправленного сектора. Таковыми являются, например, параболические антенны [3].

Следует также отметить, что разделяемый и ненаправленный характер беспроводной передающей среды обусловливает относительно низкую помехоустойчивость ФКС на ее основе. Типовое значение интенсивности битовых ошибок, BER (см. подп. 2.1.3) беспроводных ФКС составляет порядка 10^{-3}, в то время как кабельных ФКС – порядка 10^{-9} – 10^{-10} [3]. Однако данный недостаток беспроводных ФКС не является существенным благодаря широкому применению в них высокоэффективных методов помехоустойчивого кодирования, рассмотренных далее, в подп. 2.7.4 и п. 3.8.

Как было указано ранее (см. подп. 2.2.3 и 2.2.4), в качестве носителей данных беспроводных ФКС современных ВС обычно выступают электромагнитные волны частотой от сотен МГц до десятков ГГц, *основными закономерностями* распространения которых являются следующие [3]:

- *распространение* волн в однородной среде происходит по прямой линии;
- *затухание* волны в безвоздушном пространстве пренебрежимо мало, а в воздухе оно прямо пропорционально произведению квадрата частоты волны на квадрат расстояния от ее источника;
- *препятствиями* для распространения волн являются объекты с размерами, соизмеримыми с длиной волны или превышающими ее, состоящие из проводящих материалов (металла, железобетона и т. п.), в меньшей степени – из диэлектриков и полупроводников;
- при встрече волны с препятствием, геометрические размеры которого соизмеримы с ее длиной, имеет место *рассеивание* волны, то есть ее распространение по множественным траекториям, под различными углами к поверхности препятствия;
- при встрече волны с частично проницаемым для нее препятствием, размеры которого существенно превышают ее длину, происходит *отражение* волны от поверхности препятствия, под углом, равным углу падения;

- если волна встречается с непроницаемым для нее препятствием, размеры которого также существенно превышают длину волны, то при определенных условиях [3] имеет место ее *дифракция*, т. е. волна огибает препятствие;
- в ряде случаев, особенно в городских условиях, может иметь место комбинация вышеназванных эффектов.

Необходимо отметить, что явления рассеивания, отражения и дифракции волн обычно играют негативную роль при связи в микроволновом диапазоне. Они вызывают так называемое *многолучевое распространение сигнала* от передатчика к приемнику, т. е. по нескольким различным траекториям, с неодинаковым временем распространения. При этом на вход приемника поступает несколько копий одного того же сигнала, смещенных относительно друг друга по времени и, в общем случае, с различными амплитудами. Данный эффект, в свою очередь, может привести к серьезным искажениям входных сигналов приемников. Поэтому в беспроводных линиях связи, работающих в микроволновом диапазоне частот, применяются различные методы устранения или собственно указанного эффекта, или вызываемых им искажений данных. Простейшим по идее (но часто – не с точки зрения практической реализации) из этих методов является размещение антенн приемопередающей аппаратуры ФКС в пределах прямой видимости (см. подп. 2.2.3 и 2.2.4). Также широко применяются различные методы помехоустойчивого кодирования, рассмотренные далее, в подп. 2.7.4 и п. 3.8.

2.3.5. Области применения различных типов передающей среды ФКС

UTP-кабели характеризуются невысокой стоимостью, удобством прокладки и монтажа, с одной стороны, и сравнительно высоким затуханием и низкой помехоустойчивостью (см. табл. 2.1) – с другой. Поэтому обычно они применяются в наиболее «массовых» и одновременно наименее протяженных кабельных ФКС – прокладываемых внутри зданий линиях связи ЛВС протяженностью порядка единиц – десятков метров, а с недавнего времени – и в АЛС КТСОП.

STP-кабели, благодаря более высокой по сравнению с UTP помехоустойчивости при сопоставимом затухании (см. табл. 2.1) и повышенных стоимости и сложности монтажа, как правило, используются

для прокладки кабельных ФКС, требующих повышенной защиты от помех, протяженностью порядка десятков метров (в основном, внутри зданий, значительно реже – вне зданий).

ВОК отличаются малым затуханием и высокой помехоустойчивостью при относительно высокой стоимости прокладки и монтажа. Поэтому их применение технически и экономически оправдано, в первую очередь, в ФКС с высокой пропускной способностью, длиной порядка единиц – десятков километров, например, в прокладываемых вне зданий магистральных ФКС. В частности, ВОК широко применяются в МЛС КТСОП и в выделенных магистральных линиях связи ГВС, сетей мегаполисов и крупных ЛВС (см. рис. 2.6). С недавнего времени применение ВОК становится оправданным и в ФКС более «низкого ранга», вплоть до абонентских [3].

Беспроводная передающая среда применяется для реализации ФКС при сложности, невозможности или экономической невыгодности прокладки кабельных трасс, а также для связи с ВС *мобильных*, т. е. перемещающихся в пространстве абонентов (например, при поездках в транспортных средствах и т. п.). Следует отметить, что в настоящее время беспроводные ФКС получают все более широкое распространение как в ЛВС, так и в ГВС [3].

2.4. Общие вопросы представления двоичных данных в ФКС ВС

Как указано ранее (см. подп. 2.2.2), в ФКС ВС двоичные данные обычно представляются одним из трех типов сигналов-носителей: двухуровневыми, многоуровневыми или модулированными синусоидальными. В общем случае, сигналы-носители ФКС ВС, очевидно, должны удовлетворять следующим *базовым требованиям* [1, 5]:

- соответствие частотного диапазона сигнала-носителя полосе пропускания ФКС, математически выражаемое следующим образом:

$$\left.\begin{array}{l} f_{LS} \geq f_{LC}, \\ f_{HS} \leq f_{HC}, \end{array}\right\}, \qquad (2.13)$$

где f_{LS} и f_{HS} – нижняя и верхняя граничные частоты амплитудного спектра сигнала-носителя;

f_{LC} и f_{HC} – нижняя и верхняя граничные частоты полосы пропускания ФКС;

- обеспечение *самосинхронизации* передатчика и приемника, сводящееся к соблюдению следующего неравенства:

$$K_{\max} \le 0{,}5/\delta f_{\max}, \qquad (2.14)$$

где $K_{\max}$ – максимально возможное количество последовательных тактов с неизменным состоянием сигнала-носителя;

$\delta f_{\max}$ – максимально возможное относительное отклонение тактовых частот передатчика и приемника;

- для ФКС на основе электрического кабеля – также дополнительное условие обеспечения нулевой или пренебрежимо малой постоянной составляющей сигнала [3]:

$$X_{=} \approx 0. \qquad (2.15)$$

Условие (2.14) нуждается в комментарии. Очевидно, надежное считывание уровней принимаемого сигнала-носителя требует синхронизации моментов считывания с началами или с серединами *тактовых интервалов*, т. е. интервалов времени (*тактов*), в течение каждого из которых передается один бит сообщения или представляемая определенным состоянием сигнала-носителя группа битов. Следует заметить, что длительность такта не всегда совпадает с $T_{\min}$ (см. рис. 2.2), что имеет место, например, для *манчестерского кода*, описанного в подп. 2.5.5. При передаче данных на небольшие расстояния (например, между абонентами, находящимися на одной и той же печатной плате) задача синхронизации обычно решается введением между ними дополнительной линии для передачи синхроимпульсов, фронт или спад которых совпадает с серединой или (реже) с началом тактовых интервалов. Однако в ФКС ВС выделение дополнительной линии связи для синхронизации или весьма затруднительно, или (чаще) невозможно. Поэтому данная проблема может быть решена только методом *самосинхронизации*, т. е. посредством определения приемником начал или середин тактовых интервалов непосредственно из сигнала-носителя. При этом положения во времени начал и середин тактов оцениваются приемником как моменты, отстоящие от некоторого перепада сигнала-носителя (см. рис. 2.2, *а* и 2.2, *б*) на интервал времени соответственно kT и $kT + (T/2)$, где k – целое число, а T – длительность такта. Вообще говоря, корректное оценивание осуществимо только при равенстве тактовых частот передатчика и приемника. Однако, из-за неизбежного отклонения тактовых частот приемника и передатчика, с ростом k накапливается отклонение оце-

нок начал или середин тактов, определяемых в процессе самосинхронизации, от их действительных положений во времени. Поэтому серия из более $0{,}5/\delta f$ тактов с неизменным состоянием сигнала-носителя приводит к сбою синхронизации и, следовательно, некорректному обмену данными.

Тип сигнала-носителя, применяемого в той или иной разновидности ФКС, определяется в первую очередь наличием или отсутствием организационно-законодательных ограничений на полосу пропускания ФКС. При их отсутствии полоса пропускания ФКС находится в пределах от 0 Гц до некоторой верхней граничной частоты. Такой полосой пропускания характеризуются выделенные кабельные ФКС. Наиболее рациональное использование указанной полосы обеспечивается при применении двух- и многоуровневых сигналов-носителей, частотный диапазон которых находится в пределах от 0 до $1/T_{\min}$ (см. рис. 2.2 и пояснения к нему). При наличии организационно-законодательных ограничений на полосу пропускания ФКС (что имеет место в АЛС КТСОП и в беспроводных ФКС) обе ее граничные частоты отличны от нуля, и условие (2.13) может быть удовлетворено только при использовании модулированных сигналов-носителей (см., например, рис. 2.2, *в* и пояснения к нему).

Представление двоичных данных двух- и многоуровневыми сигналами называют *линейным кодированием*, а собственно эти сигналы – *линейными кодами* [1, 3, 5]. В свою очередь, представление данных модулированными сигналами называется *модуляцией*.

Известен ряд способов как линейного кодирования, так и модуляции. Большинство из этих способов не обеспечивает выполнение условий (2.14) и (2.15) для любой возможной последовательности нулей и единиц. Поэтому подлежащая передаче двоичная последовательность перед ее преобразованием в линейный код или, соответственно, перед ее использованием в качестве модулирующего сигнала часто подвергается *логическому кодированию* [1, 5], т. е. преобразованию

в другую двоичную последовательность, при представлении которой соответствующим сигналом-носителем условия (2.14) и (2.15) гарантированно *выполняются*. Во многих практических случаях логическое кодирование также обеспечивает ряд других преимуществ, например, возможность повышения помехоустойчивости обмена данными или

увеличения скорости передачи данных при ограниченной полосе пропускания ФКС [1, 5].

Обобщенная операционная модель процесса передачи двоичных данных по ФКС представлена на рис. 2.18.

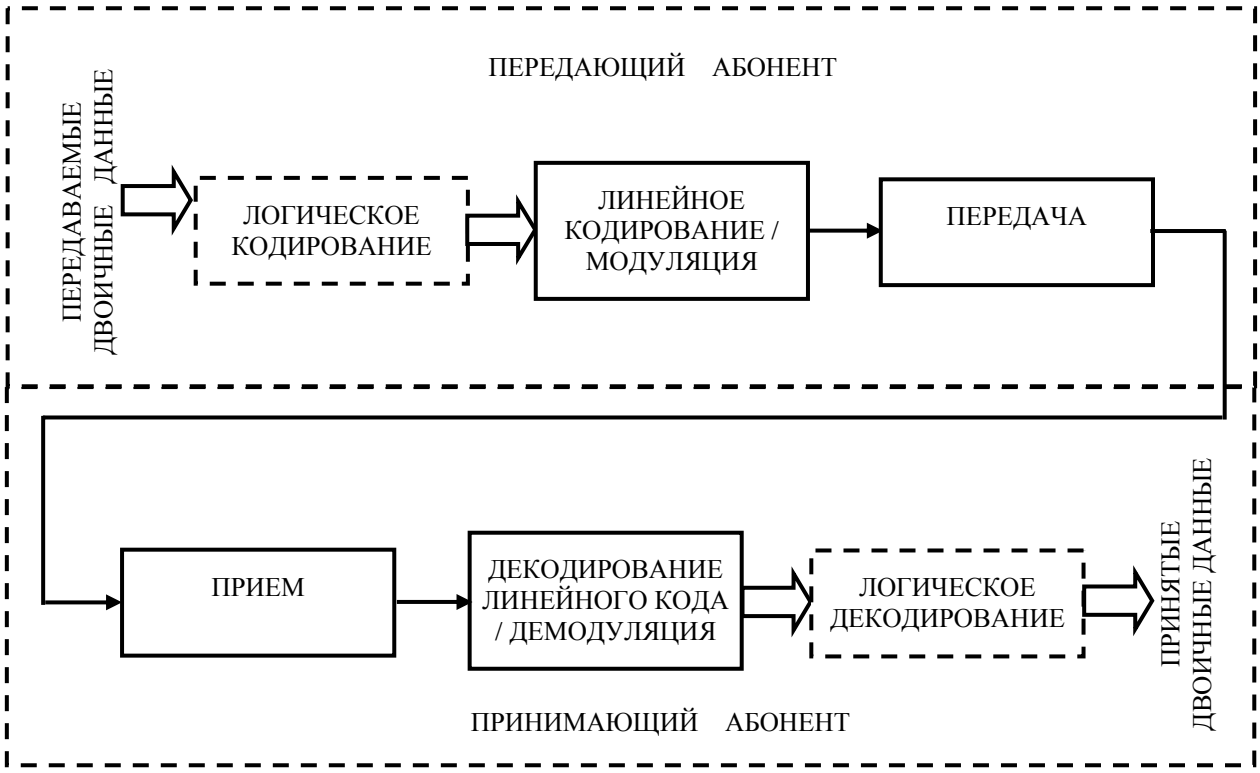


Рис. 2.18. Обобщенная операционная модель процесса передачи двоичных данных по ФКС

Необходимо также отметить, что на практике при обмене данными по ФКС между абонентами ВС часто возникает необходимость *уплотнения* (*мультиплексирования*) каналов связи [3, 5]. Оно состоит в использовании одного и того же участка передающей среды (например, отрезка кабеля) множеством ФКС. Необходимость мультиплексирования кабельных линий связи обусловлена технической и организационно-экономической сложностью прокладки отдельного участка кабеля для каждого из ФКС во многих практических случаях. Беспроводные же линии связи *всегда* разделяются множеством ФКС ввиду самой природы беспроводной передающей среды (см. подп. 2.3.4). В свою очередь, вследствие использования одного и того же участка передающей среды множеством ФКС возникает необходимость в специальных мерах для устранения взаимного влияния сигналовносителей данных этих ФКС при их передаче по общему участку среды. Указанные меры реализуются посредством специальных проце-

дур, также обычно называемых мультиплексированием и являющихся составной частью процессов модуляции (линейного кодирования) и/или логического кодирования. Простейшим примером указанных процедур является *частотное мультиплексирование* [5]. Его принцип состоит в представлении потоков данных в каждом из ФКС, разделяющих один и тот же участок передающей среды, сигналами-носителями, находящимися в различных, не перекрывающихся между собой частотных диапазонах. Естественно, кроме частотного, известны и другие методы мультиплексирования [3, 5].

Рассмотрим основные методы линейного кодирования, модуляции, логического кодирования и мультиплексирования, применяемые при представлении двоичных данных в ФКС ВС.

2.5. Линейное кодирование в ФКС ВС

Как было сказано ранее, линейным кодированием называется представление двоичных данных двух- или многоуровневыми сигналами-носителями (см. рис. 2.2, *а* и 2.2, *б*).

При представлении данных в ВКЛС ВС наибольшее распространение получили следующие типы линейных кодов [1, 3, 5]:

- двухуровневый код без возврата к нулю (*NRZ-код*);
- двухуровневый код без возврата к нулю с инверсией (*NRZI-код*);
- трехуровневый код с альтернативной инверсией (*AMI-код*);
- трехуровневый код *MLT-3*;
- двухуровневые коды группы *1B2B*;
- многоуровневые коды группы *kB1Q* (где *k* – число бит, представляемых одним уровнем линейного кода).

Примеры временных диаграмм перечисленных линейных кодов, представляющих одну и ту же двоичную последовательность, приведены на рис. 2.19. Указанные диаграммы даны в предположении, что линейные сигналы являются электрическими (напряжениями или токами) и биполярными, так как для униполярных сигналов невозможно обеспечить условие (2.15).

В ВКЛС на основе ВОК в качестве носителей данных применяются *униполярные* варианты вышеперечисленных линейных кодов, так как невозможно реализовать разнополярные импульсы излучения [3]. При этом различные уровни линейного кода представляются различ-

ными значениями мощности излучения (минимальный уровень – как правило, нулевым значением).

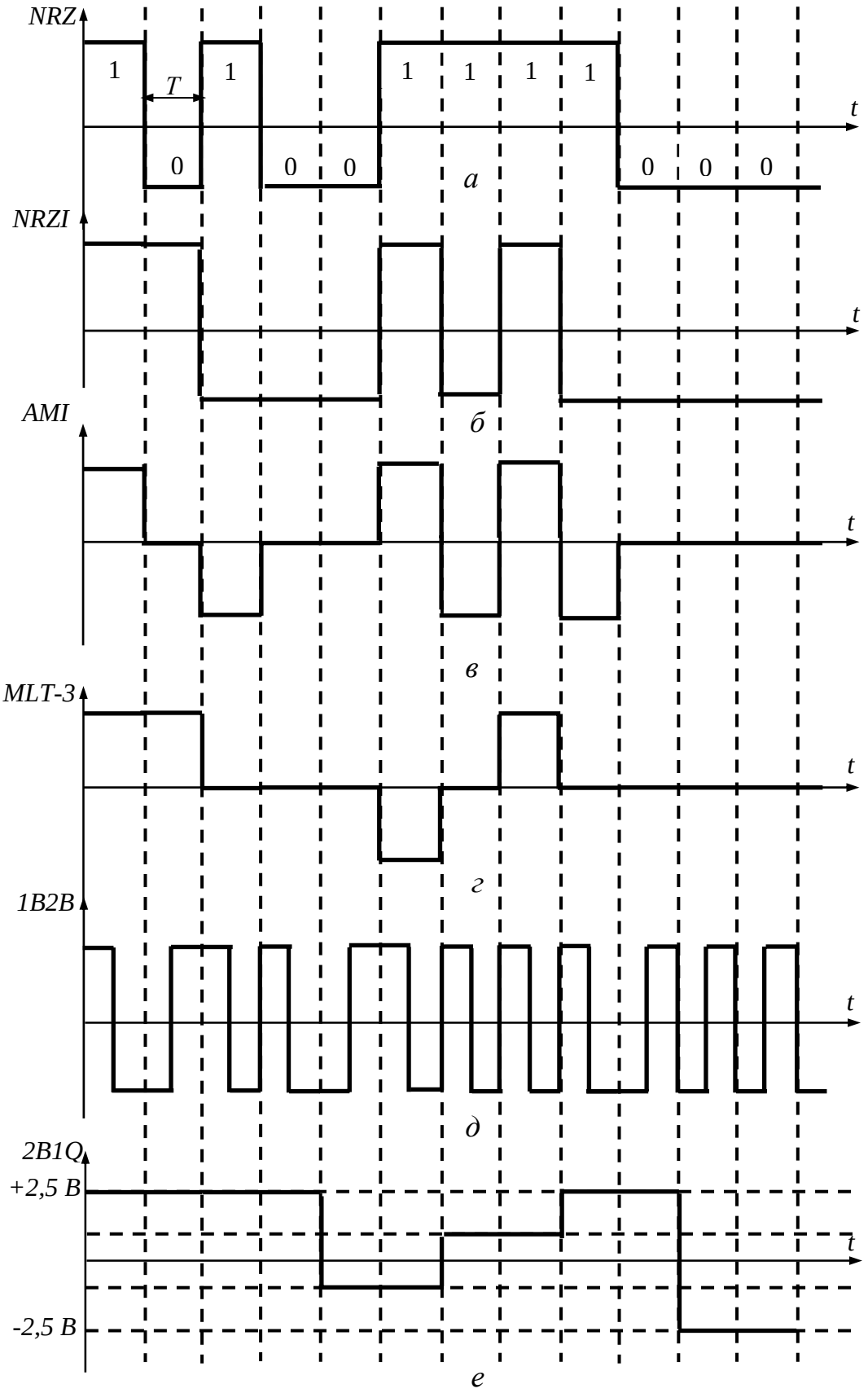


Рис. 2.19. Временные диаграммы распространенных типов линейных кодов:
а – NRZ; *б* – NRZI; *в* – AMI; *г* – MLT-3; *д* – 1B2B (Манчестер-I); *е* – 2B1Q

Необходимо отметить, что приемники излучения обычно устойчиво распознают только два уровня его интенсивности, поэтому в качестве сигналов-носителей ВОК, как правило, применяются двухуровневые линейные коды (NRZ, NRZI, 1B2B), хотя известно использование в ВОК и многоуровневых линейных кодов [1, 3, 5].

2.5.1. NRZ-код

Данный код является прямым представлением последовательности нулей и единиц некоторыми уровнями напряжения, тока или мощности излучения, например, единицы – высоким уровнем, нуля – низким (см. рис. 2.19, *а*). Название NRZ является аббревиатурой английского словосочетания «*Non-Return to Zero*» («Без возврата к нулю») и происходит от того свойства NRZ-кода, что при передаче последовательности единиц он не возвращается к нулевому уровню в течение такта, в отличие от ряда других линейных кодов (см. далее). Основные достоинства NRZ-кода – простота формирования и декодирования, а также, благодаря наличию только двух информативных уровней – сниженные требования к отношению «сигнал-шум» на линии и простота применения в ФКС на основе ВОК. Его недостатки – невозможность удовлетворения условий (2.14) и (2.15) без предварительного логического кодирования представляемой двоичной последовательности (п. 2.7).

Значение $T_{\min}$ NRZ-кода равно длительности тактового интервала T, который, в свою очередь, совпадает с интервалом передачи одного бита (см. рис. 2.2, *а* и 2.19, *а*). Поэтому, как следует из выражений (2.3) и (2.13), максимальная скорость передачи NRZ-кода по ФКС с шириной полосы пропускания Δf составляет в общем случае (т. е. при равной вероятности всех возможных двоичных комбинаций исходных данных) примерно Δf бит в секунду [5].

В целом, NRZ-кодирование уступает другим способам линейного кодирования по совокупности характеристик [3]. Поэтому применяется оно относительно редко для представления данных в ВКЛС.

2.5.2. NRZI-код

Название данного кода происходит от английского словосочетания «*Non-Return to Zero Inverted*», в переводе – «Без возврата к нулю с инверсией». Он формируется одним из следующих способов. Согласно первому из них логическая единица представляется уровнем, противоположным уровню, которым была представлена предыдущая единица, а логический ноль – совпадающим с уровнем NRZI-сигнала в предыдущем такте (см. рис. 2.19, *б*). Такой вариант NRZI-кода известен также под названием NRZ-1 (NRZ с переключением по единице). Согласно второму способу ноль представляется уровнем, противоположным таковому предыдущего нуля, а единица – совпадающим с уровнем сигнала в предыдущем такте. Данный вариант NRZI-кода известен также под названием NRZ-0 (NRZ с переключением по нулю).

Как нетрудно заметить из рис. 2.19, *б*, NRZ-1-кодирование двоичных данных позволяет устранить длинные последовательности единиц, а NRZ-0-кодирование – нулей. Однако первый из названных способов кодирования не устраняет длинных последовательностей нулей, а второй – соответственно, единиц. Кроме того, ни один из них не обеспечивает нулевой постоянной составляющей NRZI-сигнала. Поэтому обеспечение условий (2.14) и (2.15) для NRZI-кода требует предварительного логического кодирования исходной двоичной последовательности (п. 2.7). Следует отметить, что удовлетворение этих условий при NRZI-кодировании обеспечивается более простыми алгоритмами преобразования исходных данных, чем при NRZ-кодировании [1, 5].

Аналогично NRZ-коду, значение $T_{\min}$ NRZI-кода равно длительности битового интервала, а максимальная скорость передачи – в общем случае, Δf бит в секунду [5].

В целом, NRZI-код в сочетании с предварительным логическим кодированием нашел относительно широкое распространение при представлении данных в ВКЛС, в том числе (благодаря двум информативным уровням) – в волоконно-оптических линиях связи. В частности, он используется одним из стандартов технологии ЛВС Fast Ethernet, а также технологией ЛВС FDDI [3].

2.5.3. AMI-код

Название данного кода происходит от английского словосочетания *Alternate Mark Inversion*, в переводе – «Чередующаяся инверсия маркеров». Он отличается от ранее рассмотренных линейных кодов наличием не 2-х, а 3-х информативных уровней – $-X$, 0 и $+X$ (см. рис. 2.19, *в*). AMI-код формируется по следующим правилам. Логический ноль представляется нулевым уровнем, а логическая единица – уровнем, отличным от нуля, с полярностью, противоположной полярности предыдущей единицы.

Как можно увидеть из рис. 2.19, *в*, линейный код AMI характеризуется практически нулевой постоянной составляющей и эффективно устраняет длинные последовательности единиц исходного двоичного кода путем их преобразования в последовательности разнополярных импульсов. К достоинствам AMI-кода относится также возможность распознавания некоторых типов ошибочных сигналов. Так, нарушение порядка чередования полярности сигналов говорит о ложном импульсе или исчезновении с линии корректного импульса.

К основным недостаткам AMI-кода относятся следующие. Последовательность нулей при AMI-кодировании преобразуется также в последовательность нулей, т. е. тактов с неизменным (нулевым) уровнем напряжения. Следовательно, AMI-код, в общем случае, не позволяет удовлетворить условие (2.14) и требует, как и NRZ- и NRZI-коды, предварительного логического кодирования исходной двоичной последовательности. Другим существенным недостатком AMI-кода является необходимость распознавания приемником трех различных уровней сигнала (в отличие от двухуровневых линейных кодов). Дополнительный уровень требует увеличения отношения «сигнал-шум» примерно на 3 дБ для обеспечения той же достоверности приема бит на линии, как при двухуровневом линейном коде [5], а также осложняет применение AMI-кода в волоконно-оптических линиях связи.

Кроме вышеописанного базового варианта AMI-кода, существуют две его модификации, свободные от одного из перечисленных недостатков – длинных последовательностей нулей. Они известны под названиями *B8ZS-кода* (от англ. словосочетания *Bipolar with 8 Zeros Substitution*, в переводе – «Биполярный с замещениями 8 нулей»)

и *HDB3-кода* (от англ. словосочетания *High Density Bipolar 3*, в переводе – «Высокоплотный биполярный 3») [1]. Оба данных кода характеризуются следующим обобщенным алгоритмом кодирования:

- исходная двоичная последовательность преобразуется в AMI-сигнал по вышеописанным правилам (см. рис. 2.19, *в*);

- если в полученном в результате AMI-кодирования сигнале встречается непрерывная последовательность из k тактов с нулевым уровнем (где значение k равно 8-ми для B8ZS-кода и 4-м – для HDB3-кода), она заменяется некоторой другой последовательностью.

Замена осуществляется по правилам, обеспечивающим выполнение условий (2.14) и (2.15). Эти правила различны для B8ZS- и HDB3-кодов.

Правила *B8ZS-кодирования* иллюстрирует нижеприведенный рис. 2.20.

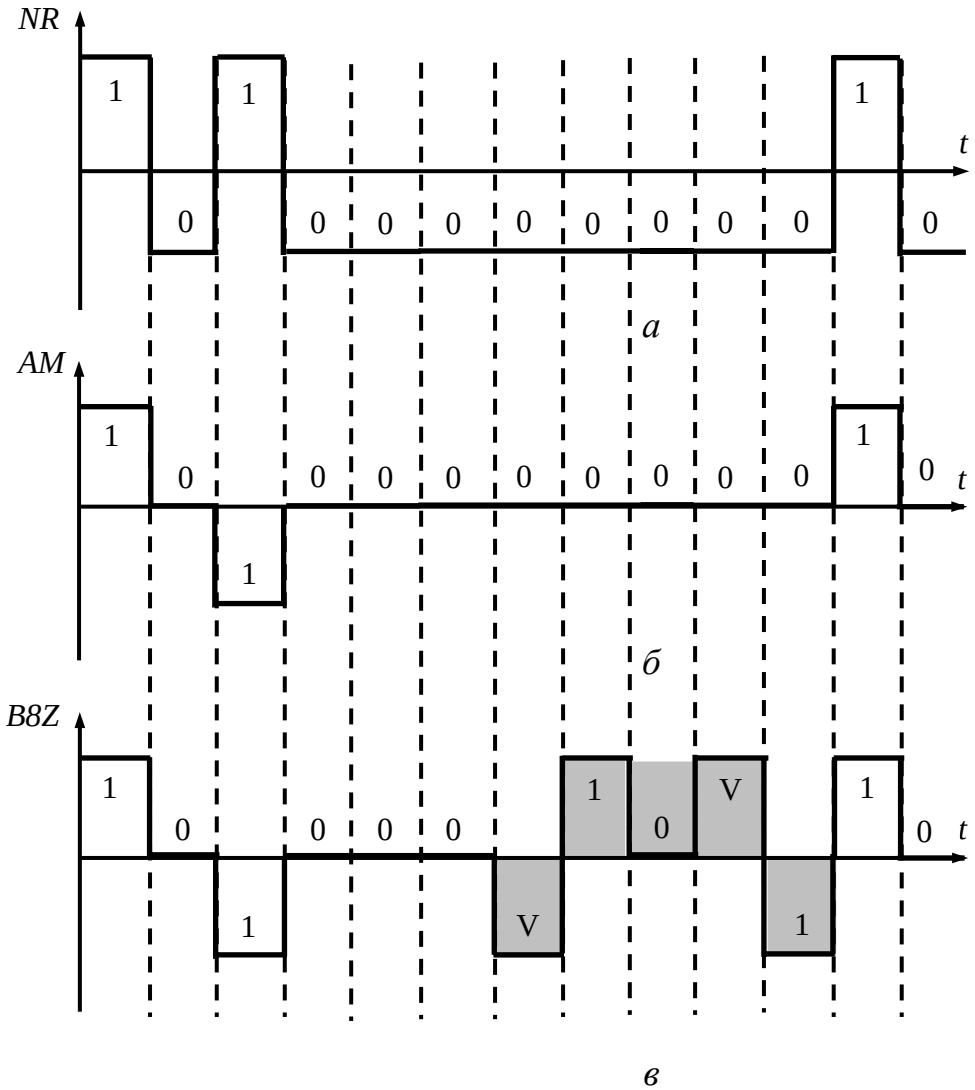


Рис. 2.20. Примеры временных диаграмм NRZ-сигнала и соответствующих ему AMI- и B8ZS-сигналов:
а – NRZ; *б* – AMI; *в* – B8ZS

Каждая встречающаяся в AMI-сигнале последовательность из 8-ми нулей при B8ZS-кодировании заменяется последовательностью 000V10V1, где:

- V (от англ. слова *violation*, в переводе – «нарушение») – импульс с полярностью, *нарушающей* правила AMI-кодирования, т. е. совпадающей с полярностью предыдущего импульса, а не противоположной ей;
- 1 – импульс с полярностью, противоположной полярности предшествующего ему V-импульса;
- 0 – такт с нулевым уровнем.

Вышеописанная подстановка обеспечивает отсутствие в B8ZS-сигнале более 7-ми тактов с неизменным (нулевым) уровнем. При этом, как нетрудно заметить из рис. 2.20, *в*:

- постоянная составляющая последовательности 000V10V1 равна нулю, т. е. замена ею 8-и нулевых тактов AMI-кода не нарушает условие (2.15);
- указанная замена не вызывает нарушений чередования полярностей импульсов исходного AMI-кода.

При B8ZS-декодировании обнаружение импульса-«нарушителя», полярность которого совпадает с полярностью предыдущего импульса, является признаком начала очередной последовательности вида V10V1, которая заменяется 5-ю нулями.

Алгоритм *HDB3-кодирования* следующий. Каждая встречающаяся в AMI-сигнале последовательность из 4-х нулей заменяется:

- последовательностью 000V при нечетном числе импульсов исходного AMI-кода с корректной полярностью, находящихся в интервале времени от ближайшего импульса - «нарушителя» (V-импульса) до заменяемой последовательности из 4-х нулей;
- последовательностью 100V при четном числе указанных импульсов.

Правила HDB3-кодирования иллюстрирует рис. 2.21.

HDB3-кодирование обеспечивает отсутствие в линейном сигнале последовательностей из более чем 3-х нулей. Также из рис. 2.21, *в* нетрудно заметить, что постоянная составляющая HDB3-сигнала прак-

тически равна нулю, однако HDB3-кодирование может потребовать изменения полярностей импульсов исходного AMI-кода.

Обнаружение замещающих 000V- и 100V-последовательностей при HDB3-детектировании, как и при B8ZS-декодировании, осуществляется по импульсам - «нарушителям».

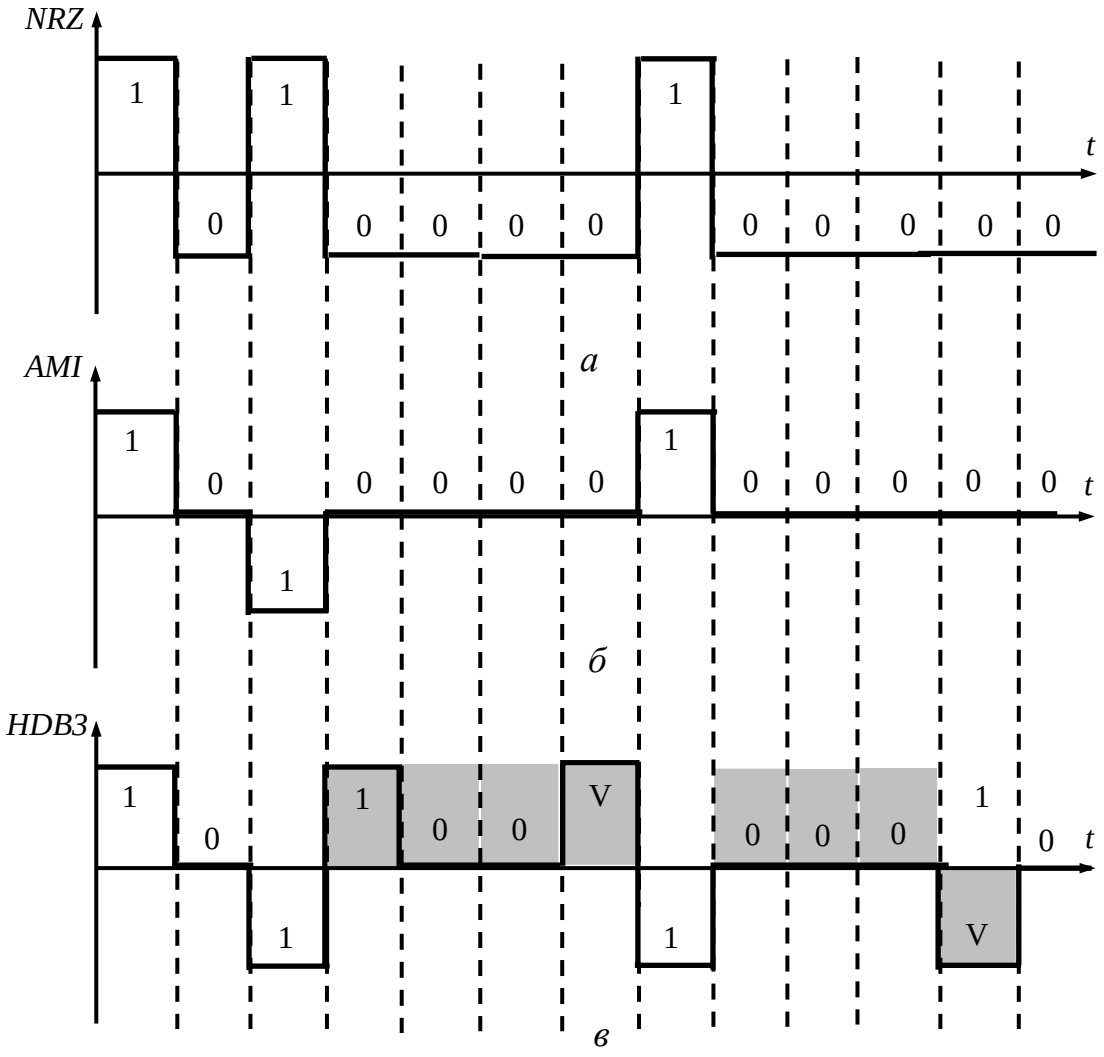


Рис. 2.21. Примеры временных диаграмм NRZ-сигнала и соответствующих ему AMI- и HDB3-сигналов: *а* – NRZ; *б* – AMI; *в* – HDB3

Значение $T_{\min}$ AMI-кода, в том числе его модификаций B8ZS и HDB3, как и ранее рассмотренных линейных кодов, равно длительности битового интервала (см. рис. 2.2 и 2.19, *в*), а максимальная скорость передачи данных, в общем случае, согласно выражениям (2.3) и (2.13) составляет примерно Δf бит в секунду [1].

Собственно AMI-код (в сочетании с предварительным логическим кодированием) применяется, например, в цифровых телефонных

сетях ISDN [3]. HDB3- и B8ZS-коды используются для представления двоичных данных в МЛС 1-го уровня иерархии КТСОП [1] (см. подп. 2.2.1) [3].

2.5.4. MLT-3-код

Название данного кода происходит от англ. словосочетания *Multi Level Transmission – 3,* в переводе – «Многоуровневая передача с 3-мя уровнями». Он, как и AMI-код, отличается 3-мя информативными уровнями - $-X$, 0 и $+X$ (см. рис. 2.19, *г*) и формируется в соответствии с табл. 2.3.

Таблица 2.3

Правила формирования MLT-3-кода

Значение текущего бита двоичной последовательности	Уровень, которым представлена предыдущая единица	Уровень, которым представлена единица, предшествовавшая предыдущей	Уровень MLT-3-кода, которым представляется текущий бит
0	Безразличен		Уровень, которым представлен предыдущий бит
1	$-X$	Безразличен	0
	0	$-X$	$+X$
	0	$+X$	$-X$
	$+X$	Безразличен	0

Как и AMI-код, MLT-3 устраняет проблему длинной последовательности единиц за счет ее преобразования в последовательность разнополярных импульсов, а также позволяет распознавать ошибки передачи сигналов, вызывающие нарушения чередования полярности импульсов. Однако, последовательность нулей исходного двоичного кода при MLT-3-кодировании преобразуется в последовательность тактов с неизменным уровнем (см. рис. 2.19, *г*). Кроме того, постоянная составляющая MLT-3-кода, в общем случае, не равна нулю. Поэтому для удовлетворения условий (2.14) и (2.15) MLT-3, как и ранее рассмотренные типы линейных кодов, требует предварительного логического кодирования исходной двоичной последовательности.

Как и у ранее рассмотренных линейных кодов, значение $T_{\min}$ MLT-3-кода равно длительности битового интервала (см. рис. 2.2 и 2.19, *г*), а максимальная скорость передачи данных, в общем случае, примерно равна Δf бит в секунду [5].

MLT-3-код с предварительным логическим кодированием исходной двоичной последовательности используется одним из стандартов технологии Fast Ethernet [3].

2.5.5. Двухуровневые логические коды группы 1B2B

Данные коды формируются путем преобразования двоичной последовательности по следующему обобщенному алгоритму [1, 5]:

- каждый битовый (тактовый) интервал разбивается на два равных по длительности подинтервала (откуда и происходит вышеприведенное условное обозначение 1B2B);
- в соответствии с правилами, определяемыми конкретным типом 1B2B-кода, задается уровень 1B2B-сигнала в каждом из подинтервалов; в общем случае данный уровень является функцией от значений текущего, предыдущего и последующего битов подвергаемой линейному кодированию двоичной последовательности.

Указанные правила формируются таким образом, чтобы удовлетворить условия (2.14) и (2.15).

Существует несколько типов линейных кодов группы 1B2B, различающихся между собой правилами формирования. Наиболее известными из них являются [1, 5]: манчестерский код, коды DMI, CMI, NEW, код Миллера. Все данные коды, удовлетворяют вышеуказанным условиям. Однако на практике наиболее широко применяется *манчестерский код*, отличающийся от других 1B2B-кодов наиболее простыми алгоритмами преобразования двоичной последовательности в линейный код и обратного преобразования. Различают две разновидности манчестерского кода: Манчестер и Манчестер-II. Первая из них характеризуется представлением логического нуля «низким» уровнем линейного сигнала в первом полутакте битового интервала и «высоким» – во втором полутакте. Логическая единица при этом представляется «высоким» уровнем в первом полутакте и «низким» – во втором. Для кода Манчестер-II характерно представление нуля

«высоким» уровнем в первом полутакте и «низким» – во втором, а единицы – «низким» уровнем в первом полутакте и «высоким» во втором. На рис. 2.19, *д* в качестве примера приведена временная диаграмма биполярного кода типа «Манчестер».

Из рис. 2.19, *д* нетрудно заметить, что манчестерский код обладает следующими основными достоинствами:

- состояние кода всегда изменяется дважды в течение такта, поэтому он принципиально не содержит длинных последовательностей нулей или единиц, и, следовательно, практически всегда удовлетворяет условию (2.14) без предварительного логического кодирования исходных данных;
- постоянная составляющая биполярного электрического манчестерского сигнала практически равна нулю и, следовательно, он удовлетворяет и условию (2.15) без предварительного логического кодирования исходной двоичной последовательности;
- код является двухуровневым, что облегчает его прием на фоне шумов, а также упрощает его применение в волоконно-оптических линиях связи (естественно, в них может использоваться только униполярный манчестерский сигнал);
- алгоритмы манчестерского кодирования и декодирования достаточно просты.

Основным недостатком манчестерского кода является в 2 раза большая верхняя граничная частота его спектра и, в общем случае, в 2 раза меньшая скорость передачи при заданной ширине полосы пропускания ФКС, чем у ранее рассмотренных линейных кодов. Данное свойство манчестерского кода обусловлено тем, что у него значение $T_{\min}$ равно *половине* длительности битового интервала (см. рис. 2.2, *а* и 2.19, *д*). Поэтому в соответствии с выражениями (2.3) и (2.13), максимальная скорость передачи манчестерского кода, в общем случае, составляет примерно $0{,}5\Delta f$ бит в секунду, т. е. в 2 раза меньше, чем у ранее рассмотренных линейных кодов. Это препятствует его применению в высокоскоростных ФКС. Однако, благодаря простоте формирования и декодирования, а также отсутствию необходимости в предварительном логическом кодировании для выполнения условий (2.14) и (2.15), манчестерский код используется низкоскоростными технологиями ЛВС, например, Ethernet 10 Мбит/с [3].

Необходимо отметить, что для самосинхронизации процесса считывания (т. е. определения приемником начал тактовых интервалов)

манчестерский код, как и другие коды группы 1В2В, требует передачи в начале каждого блока данных некоторой уникальной последовательности нулей и/или единиц (*преамбулы*), заранее оговоренной протоколом связи.

2.5.6. Многоуровневые коды группы kB1Q

Данные коды получили свое название от принципа их формирования: каждая возможная *k*-битовая комбинация исходной двоичной последовательности представляется определенным уровнем kB1Q-сигнала. На рис. 2.19, *е* приведен пример временной диаграммы сигнала, представляющего типовой пример кода группы kB1Q – потенциальный код 2B1Q [3, 5]. Он формируется следующим образом: комбинации бит 00 двоичной последовательности соответствует номинальный уровень напряжения минус 2,5 В; комбинации 01 – минус 0,833 В; комбинации 11 – плюс 0,833 В; а комбинации 10 – плюс 2,5 В.

Значение $T_{\min}$ кодов группы kB1Q, в отличие от ранее рассмотренных линейных кодов, равно длительности k битовых интервалов, т. е. kT (см. рис. 2.2, *б* и 2.19, *е*). Поэтому, в общем случае, согласно выражениям (2.3) и (2.13), верхняя граничная частота спектра этих кодов равна примерно $1/(kT)$, а максимальная скорость передачи – $k\Delta f$ бит в секунду. Таким образом, основным преимуществом этих кодов сравнению с ранее рассмотренными является минимум в k раз большая скорость обмена данными при заданной ширине полосы пропускания ФКС (по сравнению с манчестерским кодом – в $2k$ раз), что объясняется передачей состояния k битов в каждом из тактов.

К основным недостаткам кодов данной группы относятся:

- возможность присутствия длинных последовательностей тактов с неизменным уровнем от такта к такту и, в общем случае, ненулевая постоянная составляющая, что требует предварительного логического кодирования исходной двоичной последовательности для выполнения условий (2.14) и (2.15);
- необходимость распознавания приемником 2^k состояний линейного сигнала, что существенно ужесточает требования к отношению «сигнал-шум» на линии, а также осложняет применение kB1Q-кодов в волоконно-оптических линиях связи.

В целом, благодаря выигрышу в скорости обмена минимум в k раз по сравнению с другими типами линейных кодов, коды группы kB1Q (с предварительным логическим кодированием исходной двоичной последовательности) достаточно широко применяются в высокоскоростных помехозащищенных ФКС на основе электриче-

ского кабеля. В частности, пятиуровневый код PAM5, используется одним из стандартов технологии ЛВС Gigabit Ethernet, обеспечивая скорость передачи данных 250 Мбит/с по каждой из пар 4-парного STP-кабеля категории 5 (при суммарной скорости 1000 Мбит/с) [3].

Коды группы kB1Q, в частности, 2B1Q, с предварительным логическим кодированием исходной двоичной последовательности, также применяются рядом технологий группы xDSL [7] (см. подп. 2.2.1).

2.5.7. Техническая реализация логического кодирования / декодирования

Осуществляется посредством специального сетевого оборудования (сетевых адаптеров абонентских компьютеров и функционально аналогичных блоков коммутационных устройств ВС, например, коммутаторов, маршрутизаторов и т. п.), а также программного обеспечения этого оборудования. Формирование сигналов-носителей, представляющих линейные коды, как правило, осуществляется в программной форме, с последующим аппаратным преобразованием описывающей эти сигналы кодовой последовательности в напряжение, ток или уровень излучения. Извлечение из сигналов-носителей представляемых ими данных на приемной стороне обычно реализуется также в программной форме, после предварительного аналого-цифрового преобразования этих сигналов и, при необходимости, программной цифровой фильтрации результатов преобразования с целью подавления его неинформативных составляющих.

Поскольку сетевые адаптеры и аналогичные им блоки коммутационных устройств ВС, кроме функций физического уровня модели OSI, реализуют также и функции канального уровня, принципы их реализации будут рассмотрены далее, в гл. 3.

Выводы по п. 2.5

В нижеприведенной сводной табл. 2.4 представлены основные характеристики рассмотренных выше линейных кодов [1, 3, 5].

Необходимо отметить, что упомянутые в тексте настоящего пункта и в табл. 2.4 технологии ЛВС Ethernet, Fast Ethernet, Gigabit Ethernet, FDDI будут подробнее рассмотрены далее, в гл. 3.

В заключение следует заметить, что, как следует из вышеизложенного материала и из табл. 2.4, большинство из распространенных на практике линейных кодов требует предварительного *логического кодирования* исходной двоичной последовательности, основные методы которого будут рассмотрены далее, в п. 2.7.

Таблица 2.4

Базовые характеристики линейных кодов распространенных типов

Тип линейного кода	Число уровней	Необходимость логического кодирования исходных данных для выполнения условий (2.14) и (2.15)	Максимальная скорость обмена данными (в общем случае), бит/с	Примеры применения
NRZI	2	Есть	Δf *	Fast Ethernet; FDDI
AMI (базовый вариант)	3	Есть		ISDN
HDB3; B8ZS		Нет		МЛС КТСОП
MLT-3		Есть		Fast Ethernet
Манчестер	2	Нет	$0{,}5\Delta f$ *	Ethernet 10 Мбит/с; Token Ring
kB1Q	2^k	Есть	$k\Delta f$ *	Gigabit Ethernet; ряд xDSL-технологий

* Δf – ширина полосы пропускания ФКС

2.6. Модуляция в ФКС ВС

Модуляция, как указано в п. 2.4, применяется для представления двоичных данных в ФКС ВС при наличии организационно-законодательных ограничений на полосу пропускания ФКС и, как следствие, ненулевом значении ее нижней граничной частоты.

В общем, под модуляцией в теории и практике телекоммуникаций понимают изменение во времени какого-либо из параметров сигнала некоторой формы, обычно синусоидального, (например, амплитуды, частоты или фазы), как функции от передаваемых данных. При этом указанный сигнал называется *несущей*, собственно передаваемые данные – *модулирующим сигналом*, а несущая, некоторый параметр кото-

рой изменяется во времени как функция от передаваемых данных – *модулированным сигналом* (например, амплитудно-модулированным, частотно-модулированным и т. п.). При цифровом модулирующем сигнале и, соответственно, дискретном характере изменения информативных параметров модулированного сигнала модуляцию называют *манипуляцией* (например, амплитудной манипуляцией, частотной манипуляцией и т. п.). Извлечение данных из модулированного сигнала на приемной стороне осуществляется посредством процедуры, называемой *демодуляцией (детектированием)*.

В качестве несущей обычно применяется синусоидальный сигнал. При импульсно-кодовой модуляции (подп. 2.6.5) несущая представляет собой, по существу, последовательность прямоугольных импульсов, модулируемых по амплитуде. Известны и другие типы несущей [5], однако они не нашли широкого применения в ФКС ВС. Поэтому рассмотрение методов модуляции, основанных на их использовании, выходит за рамки настоящего учебного пособия. Частота несущей на практике обычно намного выше частоты модулирующего сигнала [5].

В качестве модулирующего сигнала при представлении двоичных данных выступает NRZ-сигнал (см. рис. 2.19, *а*), соответствующий или непосредственно исходной двоичной последовательности, или данной последовательности, подвергнутой предварительному логическому кодированию (п. 2.7).

Рассмотрим вкратце основные методы модуляции (манипуляции), применимые для представления двоичных данных в ФКС ВС. При этом будет предполагаться, что, если не оговаривается иное, несущая модулированного сигнала является *синусоидальной*.

2.6.1. Амплитудная модуляция (манипуляция)

Данный метод модуляции является наиболее давно применяемым с исторической точки зрения и наиболее простым в реализации. Амплитудно-модулированный сигнал (АМ-сигнал) характеризуется функциональной зависимостью (обычно линейной) его амплитуды от модулирующего сигнала. В общем случае, уравнение амплитудной модуляции синусоидальной несущей имеет вид [1]:

$$X_{AM}(t) = X_{m0} \times \left[1 + K_{AM} \times X_M(t)\right] \times \sin\left(\omega_0 t + \varphi_0\right), \qquad (2.16)$$

где $X_{AM}(t)$ – АМ-сигнал;

X_{m0}, ω_0 и φ_0 – соответственно амплитуда, угловая частота и начальная фаза несущей;

$X_M(t)$ – модулирующий сигнал;

K_{AM} – коэффициент амплитудной модуляции ($0 < K_{AM} \leq 1$).

При конечном числе различимых уровней модулирующего сигнала, в частности, при цифровом двухуровневом модулирующем сигнале амплитудная модуляция вырождается в амплитудную манипуляцию, обозначаемую аббревиатурой *ASK* (от англ. словосочетания *Amplitude-Shift Keying*, в дословном переводе – «Переключение со смещением по амплитуде»). Принцип ASK состоит в представлении каждого из различимых уровней модулирующего сигнала определенным диапазоном значений амплитуды синусоидальной несущей. Например, при двухуровневом модулирующем сигнале одному из уровней (например, «высокому») соответствует значение амплитуды выше некоторого порогового уровня, а другому – ниже некоторого другого порогового уровня. На рис. 2.22 приведен пример временной диаграммы ASK-сигнала, представляющего собой синусоиду, амплитудно-модулированную в соответствии с выражением (2.16) (при K_{AM}, равном 0,8) некоторой последовательностью нулей и единиц, также приведенной на рис. 2.22.

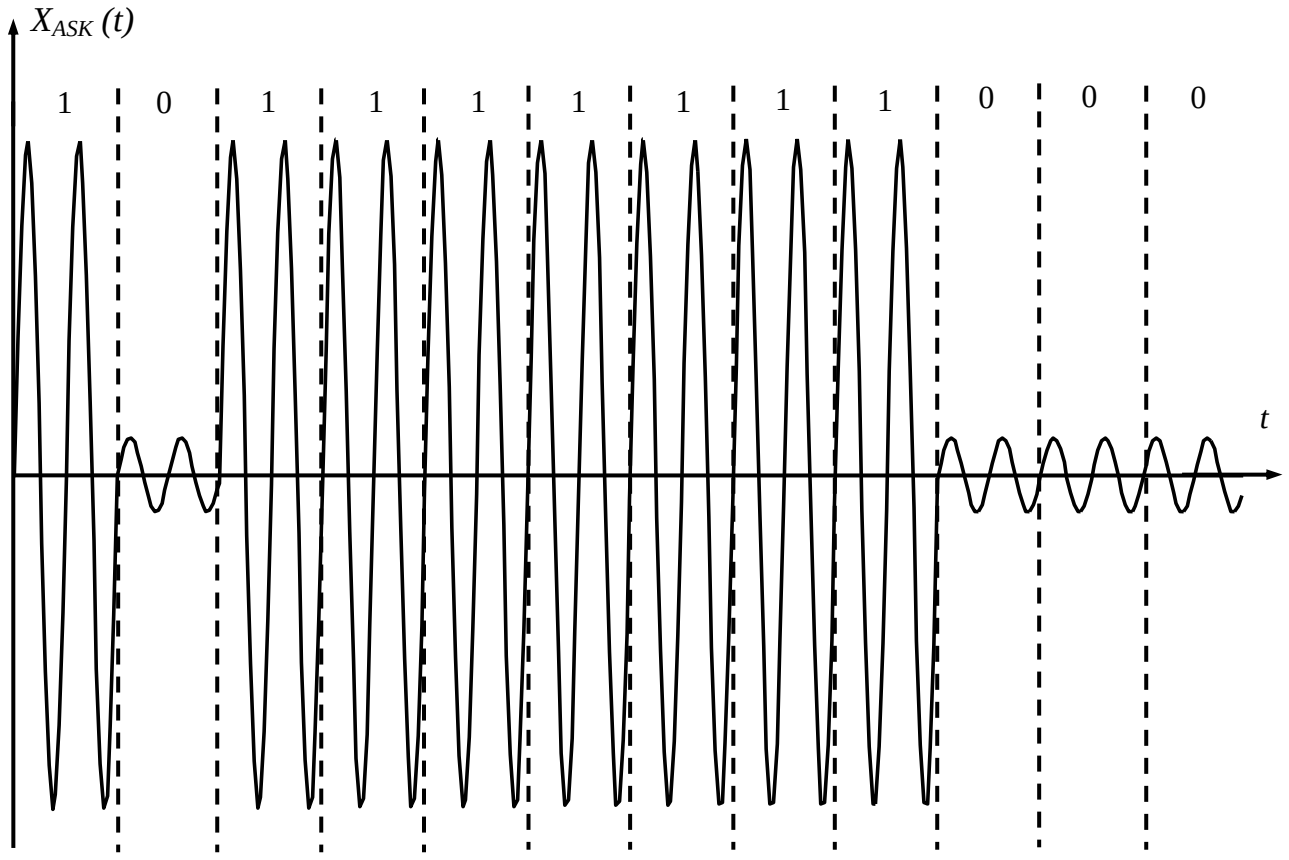


Рис. 2.22. Пример временной диаграммы ASK-сигнала

Известна и m-уровневая ASK с двоичным модулирующим сигналом, при которой каждый из m различимых диапазонов амплитуды ASK-сигнала соответствует определенной битовой комбинации модулирующего сигнала разрядностью $\log_2 m$.

Из всех методов модуляции ASK наиболее проста в реализации. Однако применение ASK в ВС весьма затруднено ввиду того, что из трех параметров синусоидального сигнала его амплитуда наиболее подвержена искажениям в каналах связи, обусловленным как помехами, так и неизбежной нестабильностью коэффициентов передачи ФКС в процессе работы ВС [1]. Поэтому амплитудная манипуляция обычно не обеспечивает приемлемой надежности обмена данными между абонентами ВС и практически не находит применения в ФКС ВС.

2.6.2. Частотная модуляция (манипуляция)

Частотно-модулированный сигнал (ЧМ-сигнал) характеризуется функциональной зависимостью его мгновенной частоты от модулирующего сигнала. Частотная модуляция синусоидальной несущей, в общем случае, описывается следующим уравнением [1]:

$$X_{FM}(t) = X_{m0} \times \sin\left\{\int_0^t \left[\omega_0 + K_{FM} \times X_M(t)\right]dt + \varphi_0\right\}, \qquad (2.17)$$

где $X_{FM}(t)$ – ЧМ-сигнал;

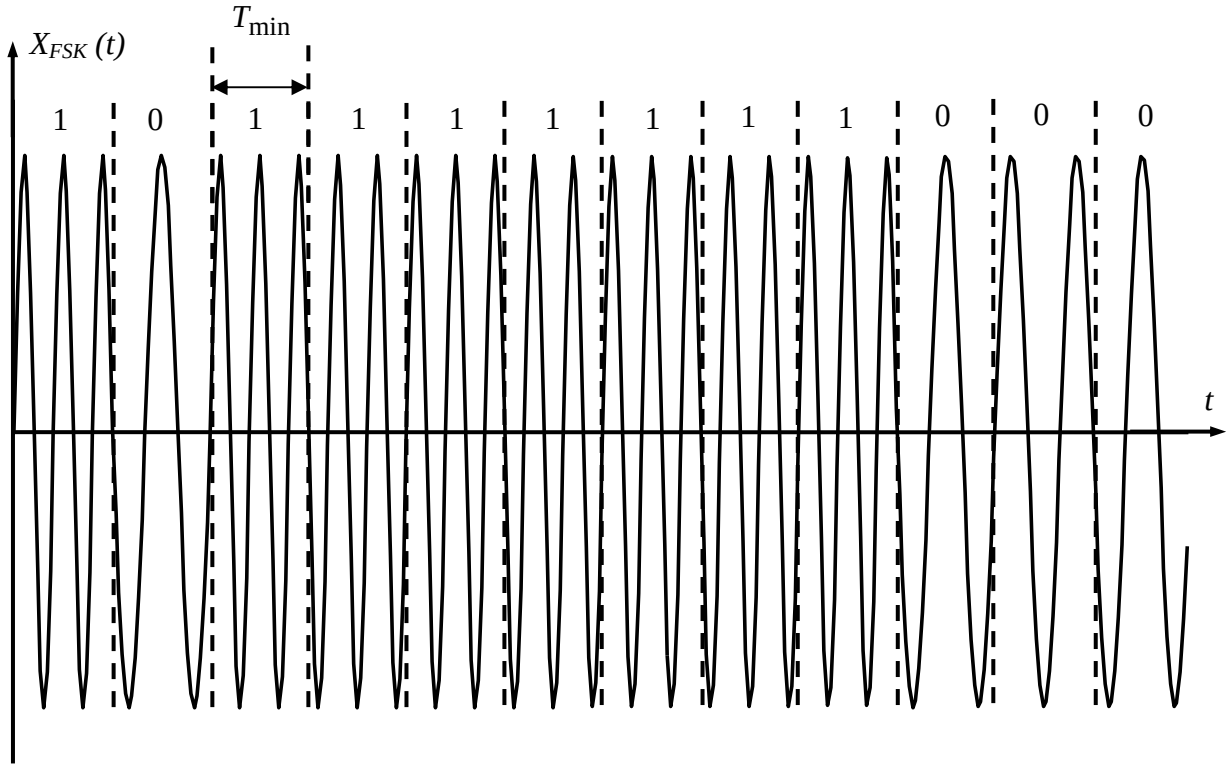
K_{FM} – коэффициент, имеющий размерность рад$^{-1}$/В или рад$^{-1}$/А;

остальные условные обозначения аналогичны таким же в уравнении (2.16).

Подынтегральное выражение в уравнении (2.17) представляет собой, по существу, мгновенную частоту ЧМ-сигнала. При конечном числе различимых уровней модулирующего сигнала частотная модуляция называется частотной манипуляцией, *FSK* (от англ. словосочетания *Frequency-Shift Keying*, в переводе – «Переключение со смещением по частоте»). Принцип FSK состоит в представлении каждого из различимых уровней модулирующего сигнала синусоидой определенной частоты (при не зависящей от него амплитуде). Известна и так

называемая m-частотная FSK [1] с двоичным модулирующим сигналом, при которой каждой из m частот FSK-сигнала соответствует определенная битовая комбинация модулирующего сигнала разрядностью $\log_2 m$. Однако на практике наиболее распространена двухчастотная FSK [1].

На рис. 2.23 представлен пример временной диаграммы двухчастотного FSK-сигнала. Он представляет собой синусоиду, частотно-модулированную двоичной последовательностью, также приведенной на рис. 2.23.



$T_{\min}$ – минимальная длительность интервала манипуляции
(интервала времени между изменениями частоты FSK-сигнала)

Рис. 2.23. Пример временной диаграммы FSK-сигнала

Граничные частоты *спектра* FSK-сигнала могут быть оценены по следующим приближенным выражениям [1]:

$$\left.\begin{aligned} f_{LS} &\approx f_0 - \frac{M+1}{T_{\min}}; \\ f_{HS} &\approx f_0 + \frac{M+1}{T_{\min}}; \end{aligned}\right\}, \qquad (2.18)$$

где M – *индекс модуляции*, для FSK-сигнала определяемый как отношение:

$$M = \Delta f_m \times T_{\min}, \qquad (2.19)$$

где Δf_m – *девиация частоты* FSK-сигнала, равная максимальному абсолютному отклонению его мгновенной частоты от частоты несущей в процессе частотной манипуляции.

Из выражений (2.18) нетрудно увидеть, что ширина спектра FSK-сигнала увеличивается с ростом индекса модуляции. При *узкополосной* частотной манипуляции, характеризуемой значением данного индекса, много меньшим единицы (иными словами – девиацией частоты, много меньшей значения $1/T_{\min}$), граничные частоты спектра FSK-сигнала стремятся к значениям, определяемым выражениями (2.4), сравниваясь с ними при M, равном нулю. При *широкополосной* частотной манипуляции, характеризуемой M, много большим единицы, граничные частоты спектра FSK-сигнала равны примерно $f_0 - (M/T_{\min})$ и $f_0 + (M/T_{\min})$ соответственно. Следовательно, в соответствии с выражениями (2.13), при заданной ширине полосы пропускания ФКС узкополосная частотная манипуляция позволяет обеспечить скорость обмена данными, в M раз большую, чем широкополосная с индексом модуляции M.

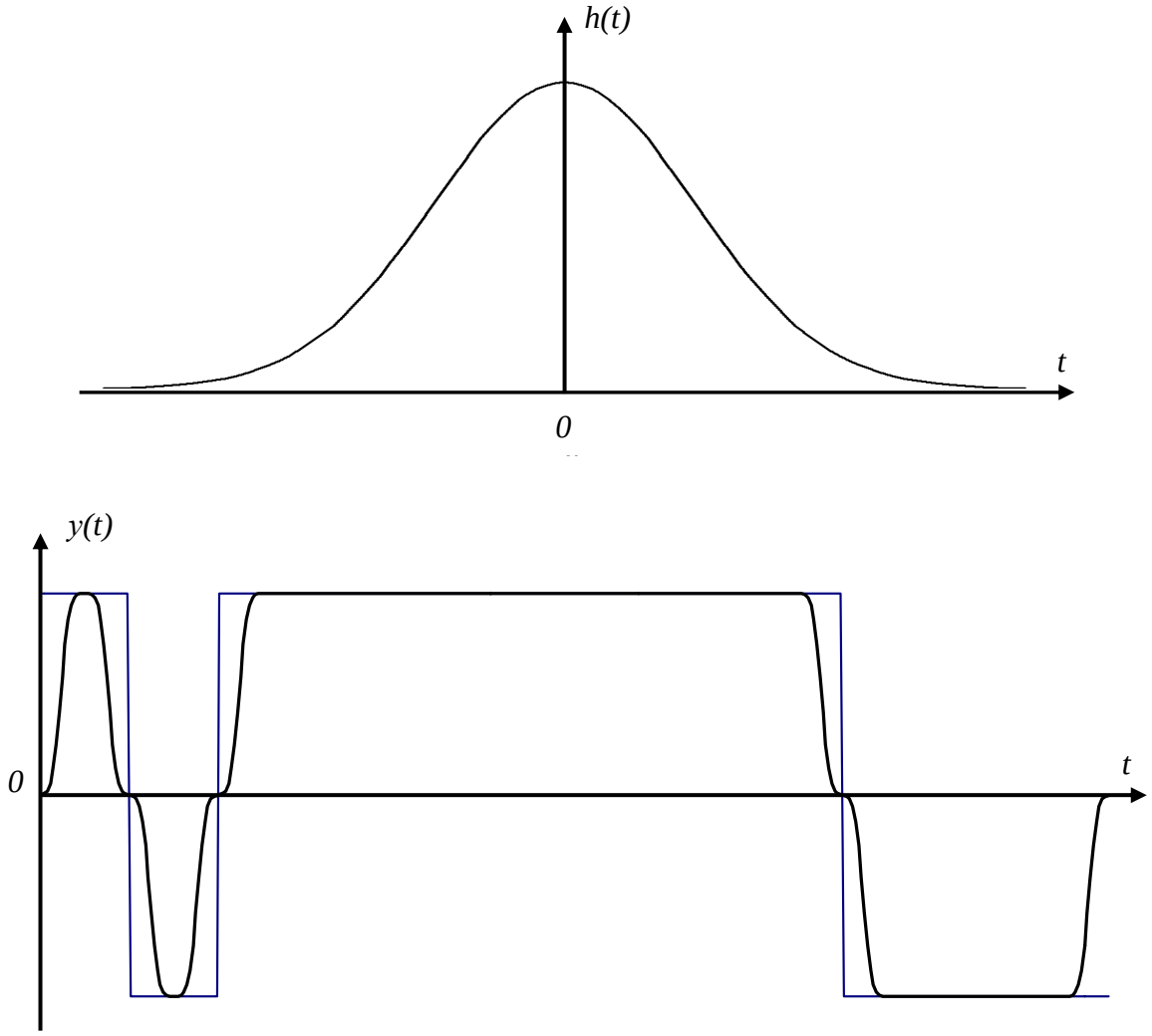
Основным недостатком узкополосной частотной манипуляции по сравнению с широкополосной являются относительно малая амплитуда изменений информативного параметра (частоты) FSK-сигнала и, соответственно, сложность их обнаружения (детектирования). Однако ввиду возможности применения в современных АПД ФКС эффективных алгоритмов цифрового детектирования указанный недостаток не является существенным. Поэтому в ФКС ВС предпочтительно применение *узкополосной* манипуляции, теоретически – с возможно меньшим индексом. Существует минимальное значение индекса модуляции, при котором возможно корректное детектирование. Показано, что оно равно 0,25 [1]. FSK с таким значением индекса известна под названием *минимальной частотной манипуляции*, *MSK* (от англ. словосочетания *Minimum Shift Keying*, в переводе – «переключение
с минимальным сдвигом»).

Также необходимо указать, что с целью сужения частотного диапазона модулированного сигнала при MSK часто используется предварительная *Гауссова* фильтрация модулирующего NRZ-сигнала [1]. Она состоит в преобразовании данного сигнала в последовательность

импульсов с «затянутыми» фронтами, имеющими форму восходящей или, соответственно, нисходящей ветви Гауссовой (колоколообразной) кривой. Это устраняет резкие перепады частоты MSK-сигнала и, как следствие, сужает его частотный диапазон, приближая его границы к значениям, определяемым выражениями (2.4). Указанное преобразование реализуется путем фильтрации модулирующего NRZ-сигнала цифровым фильтром с конечной импульсной характеристикой (КИХ), имеющей форму Гауссовой кривой (рис. 2.24, *а*). На рис. 2.24, *б* представлены примеры временных диаграмм NRZ-сигнала до и после Гауссовой фильтрации.

MSK с предварительной Гауссовой модуляцией известна в литературе под аббревиатурой *GMSK* (от англ. словосочетания *Gaussian Minimum Shift Keying*) [1].

В соответствии с выражениями (2.13) и (2.18), максимальная скорость передачи данных методом узкополосной двухчастотной FSK (MSK) по ФКС с шириной полосы пропускания Δf, в общем случае, равна примерно $\Delta f / 2$ бит в секунду.



б

Рис. 2.24. Пояснения принципа Гауссовой фильтрации: импульсная характеристика Гауссова КИХ-фильтра (*а*) и примеры временных диаграмм NRZ-сигнала до (менее интенсивная линия) и после Гауссовой фильтрации (*б*)

Информативный параметр FSK-сигнала, частота, значительно меньше подвержен искажениям в каналах связи, чем амплитуда [1]. Поэтому метод FSK (в том числе MSK), в отличие от ASK, обеспечивает высокую (максимальную среди всех методов манипуляции) надежность и помехоустойчивость обмена данными между абонентами ФКС.

Узкополосная FSK (в том числе MSK) достаточно широко применяется для передачи двоичных данных как по проводным, так и по беспроводным ФКС ВС. Ввиду наивысшей помехоустойчивости среди всех методов манипуляции (подп. 2.6.7), их использование оправдано, в первую очередь, при высоком уровне помех в ФКС, что имеет

место, например, в беспроводных ФКС, а также в ряде практических случаев – в кабельных АЛС КТСОП. В частности, применение узкополосной FSK предусматривается протоколом модуляции V.21, ориентированным на использование в зашумленных АЛС КТСОП. Использование GMSK оговаривается одним из стандартов группы IEEE 802.11 (Wi-Fi), а также рядом стандартов группы GSM [1, 3, 5].

Например, протокол V.21 оговаривает следующие характеристики FSK-сигналов [1]:

- частота несущей вызывающего абонента – 1080 Гц;
- частота несущей отвечающего абонента – 1750 Гц;
- девиация частоты – 100 Гц;
- частота FSK-сигнала, соответствующего логическому нулю, равна частоте несущей плюс 100 Гц;
- частота FSK-сигнала, соответствующего логической единице, равна частоте несущей минус 100 Гц;
- скорость передачи данных – 300 бит/с;
- ширина спектра FSK-сигнала – примерно 600 Гц.

2.6.3. Фазовая модуляция (манипуляция)

Фазово-модулированный сигнал (ФМ-сигнал) характеризуется функциональной зависимостью его фазы (строго говоря, начальной фазы) от модулирующего сигнала. Уравнение фазовой модуляции синусоидальной несущей, в общем случае, имеет вид [1]:

$$X_{PM}(t) = X_{m0} \times \sin\{\omega_0 t + \varphi_0 + K_{PM} \times X_M(t)\}, \quad (2.20)$$

где $X_{PM}(t)$ – ФМ-сигнал;

K_{PM} – коэффициент, имеющий размерность рад/В или рад/А.

Остальные условные обозначения аналогичны таким же в уравнениях (2.16) и (2.17). Необходимо отметить, что фазовая модуляция вызывает изменение также и мгновенной частоты ФМ-сигнала, как и частотная – фазы ЧМ-сигнала [1]. Поэтому в литературе оба данных вида модуляции известны под обобщенным названием *угловой модуляции***.**

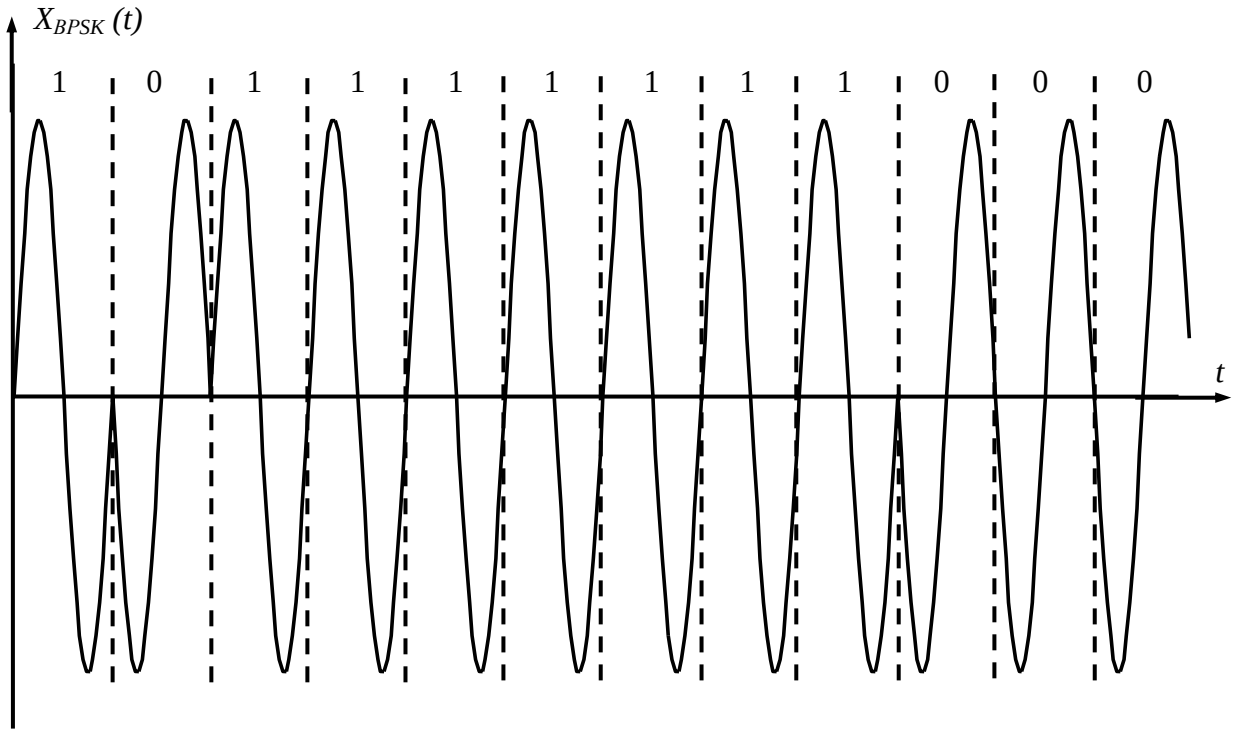
В системах связи ВС применяется фазовая модуляция с цифровым модулирующим сигналом, т. е. фазовая манипуляция, обозначаемая аббревиатурой *PSK* (от англ. словосочетания *Phase Shift Keying*, в переводе – «переключение с фазовым сдвигом»). Ее распространенной

разновидностью является *дифференциальная (относительная) фазовая манипуляция, DPSK*. В общем случае, PSK (DPSK) является m-арной, аналогично m-арной FSK (см. подп. 2.6.2).

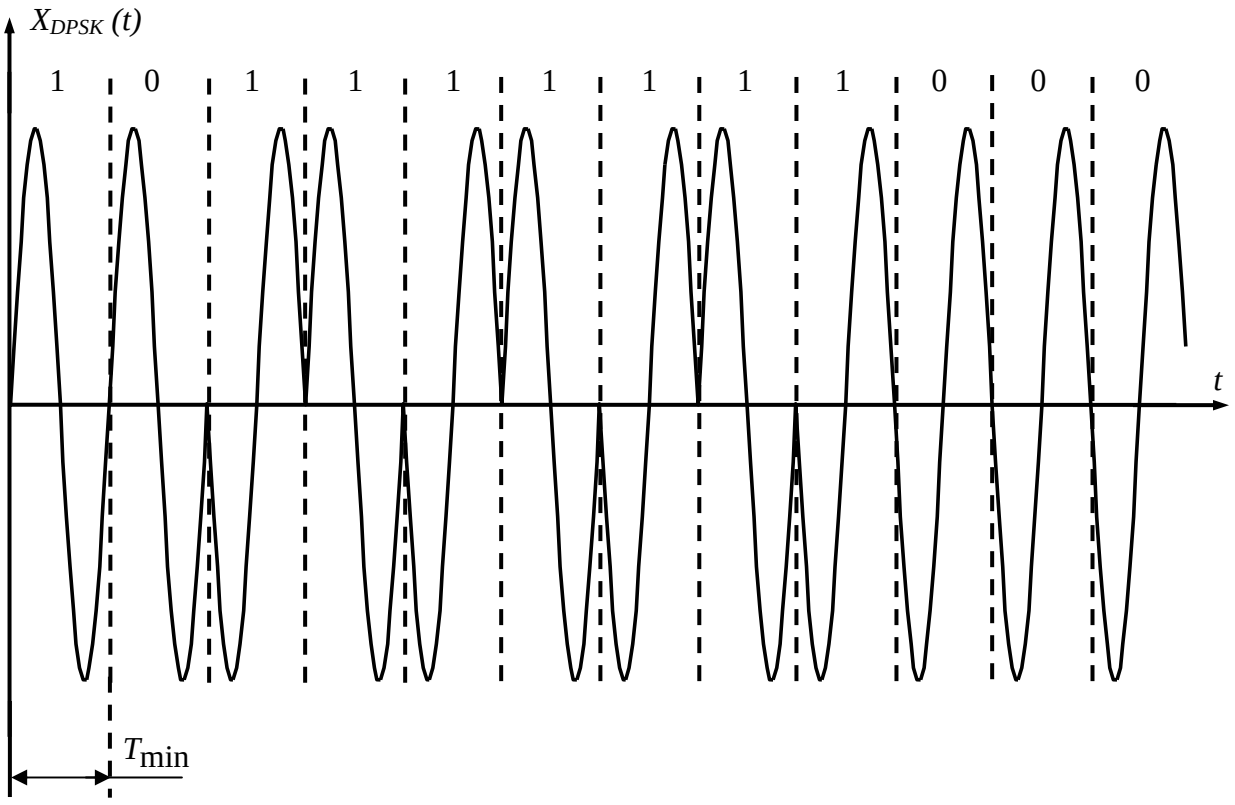
Принцип N-арной PSK состоит в том, что каждая из m возможных битовых комбинаций разрядностью $\log_2 N$ модулирующего NRZ-сигнала представляется синусоидой с определенным значением фазового сдвига относительно некоторого опорного синусоидального сигнала (в качестве которого обычно выступает несущая). Как пример, рассмотрим простейший случай PSK – двоичную PSK (*BPSK*, от англ. словосочетания *Binary Phase Shift Keying*). Она характеризуется параметром N, равным 2-м, и, соответственно, 2-мя возможными значениями фазового сдвига PSK-сигнала, при разрядности кодируемой ими битовой комбинации, равной 1 бит. Двоичный ноль модулирующего NRZ-сигнала при этом представляется, например, синусоидой с нулевым фазовым сдвигом относительно несущей, а двоичная единица – синусоидой, сдвинутой по фазе относительно несущей на 180°. На рис. 2.25, *а* приведен пример временной диаграммы BPSK-сигнала, представляющего некоторую, приведенную там же двоичную последовательность.

На рис. 2.25, *б* приведен результат бинарной DPSK двоичной последовательности, также представленной на рис. 2.25, *б* и совпадающей с последовательностью, соответствующей приведенному на рис. 2.25, *а* BPSK-сигналу.

Детектирование DPSK-сигнала проще в реализации, чем детектирование PSK-сигнала, и более устойчиво к нестабильности как параметров детектора, так и фазы модулированного сигнала.



а



$T_{\min}$ – минимальная длительность интервала манипуляции (интервала времени между изменениями фазы PSK-сигнала)

Рис. 2.25. Примеры временных диаграмм PSK-сигнала (*а*) и DPSK-сигнала (*б*)

Это обусловлено отсчетом фазы детектируемого сигнала относительно того же сигнала в предыдущем такте, а не относительно несущей, что, в свою очередь, не требует выделения несущей в процессе детектирования, а также долговременной стабильности параметров детектора и начальной фазы несущей. Благодаря указанным преимуществам, DPSK более распространена на практике, чем PSK [1, 5].

Основным недостатком DPSK, по сравнению с PSK, является *распространение ошибки*, т. е. влияние ошибки определения фазового сдвига в некотором такте на все последующие такты. Однако данный эффект может быть устранен *помехоустойчивым кодированием* модулирующего цифрового сигнала (подп. 2.7.4 и п. 3.8).

Необходимо отметить, что PSK (DPSK), аналогично Манчестерскому кодированию (см. подп. 2.5.5), для корректного детектирования требует передачи в начале каждого блока данных некоторой *преамбулы*, т. е. представляемой PSK-сигналом некоторой заранее оговоренной протоколом связи последовательности нулей и/или единиц, используемой для настройки детектора на начала тактовых интервалов.

Кроме вышерассмотренных PSK (DPSK) с N, равным 2-м, на практике достаточно широко применяются PSK (DPSK) с N, равным 4-м. Они известны под обобщенным названием *квадратурной фазовой манипуляции, QPSK* (от англ. словосочетания *Quadrature Phase Shift Keying*). Дифференциальная QPSK обычно обозначается аббревиатурой DQPSK. Применительно к ней в отечественной литературе используется также термин *двойная относительная фазовая манипуляция (ДОФМ)*.

QPSK (DQPSK) характеризуется применением не 2-х, а 4-х значений фазового сдвига модулированного сигнала относительно несущей (PSK) или того же сигнала в предыдущем интервале манипуляции (DPSK) для представления цифровых данных. При этом разрядность битовой комбинации, представляемой каждым из указанных 4-х значений фазового сдвига, равна 2-м битам.

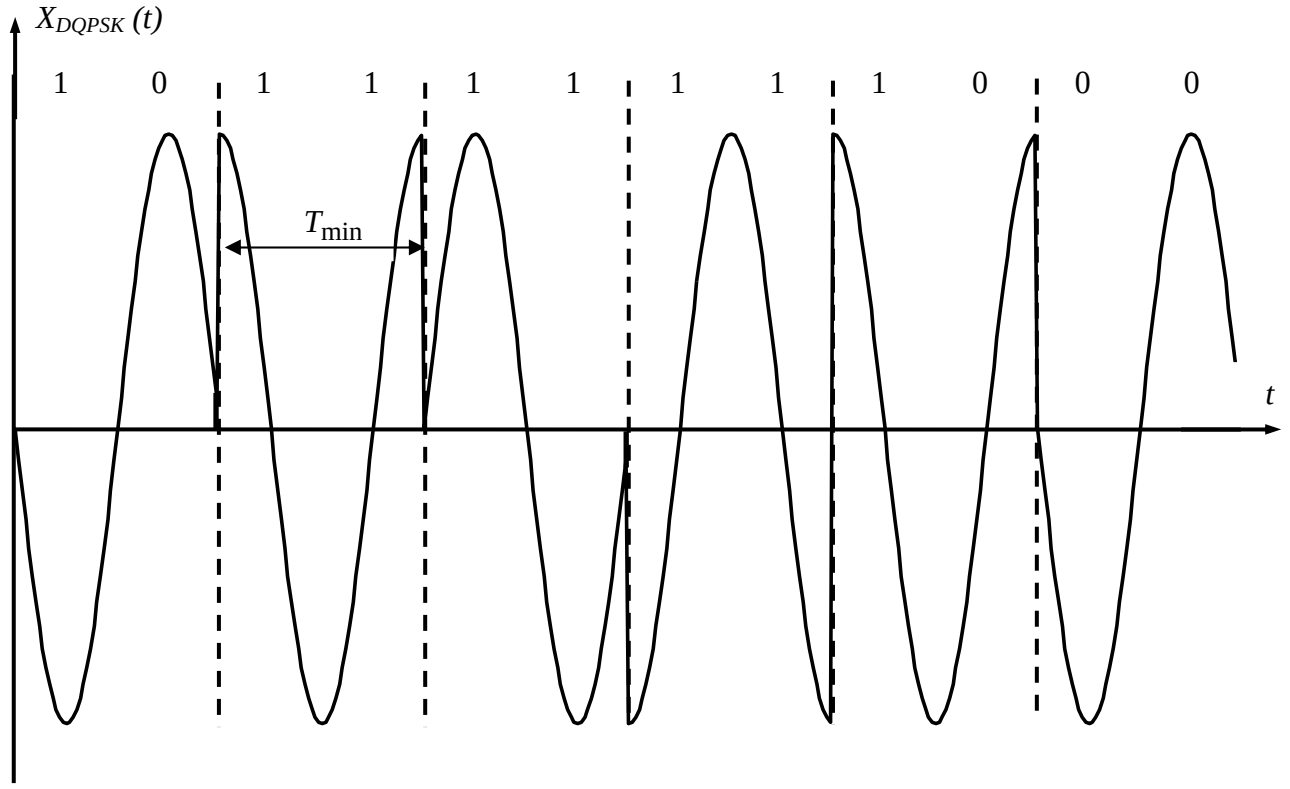
При DQPSK обычно применяются значения фазовых сдвигов, равные 0°, 90°, 180° и 270° или 45°, 135°, 225° и 315°. Например, протокол модуляции V.22, использующий DQPSK, предполагает кодирование двухбитовых двоичных комбинаций (*дибитов*) в соответствии с нижеприведенной табл. 2.5.

Таблица 2.5

Представление дибитов согласно протоколу V.22

Значение дибита	Фазовый сдвиг DPSK-сигнала относительно того же сигнала в предыдущем интервале манипуляции
00	90°
01	0°
11	270°
10	180°

На рис. 2.26 приведен пример временной диаграммы DQPSK-сигнала, представляющей (в соответствии с табл. 2.5) двоичную последовательность, также приведенную на рис. 2.26.



$T_{\min}$ – минимальная длительность интервала манипуляции (интервала времени между изменениями фазы DQPSK-сигнала)

Рис. 2.26. Пример временной диаграммы DQPSK-сигнала

Следует отметить, что на практике часто применяются дополнительные приемы кодирования дибитов фазовыми сдвигами, смысл которых состоит в устранении резких перепадов фазы и, соответственно, амплитуды модулированного сигнала на границах между тактовыми интервалами (см. рис. 2.24). Это, в свою очередь, снижает ширину спектра модулированного сигнала [5].

В принципе, известна PSK (DPSK) с N, равным 8-ми и, соответственно, разрядностью битовой комбинации, передаваемой в каждом тактовом интервале, равной 3-м. Для кодирования трехбитовых двоичных комбинаций от *000* до *111* при этом используются 8 различающихся между собой на 45° значений фазового сдвига модулированного сигнала относительно несущей (PSK) или того же сигнала в предыдущем такте (DPSK). Очевидно, при заданной ширине полосы пропускания канала связи данный тип PSK отличается в 1,5 раза большей скоростью передачи, чем QPSK (DQPSK) и в три раза большей – чем BPSK (DBPSK). Однако PSK (DPSK) с N, равным (а также большим) 8-ми, не нашла широкого применения на практике, так как уступает другим используемым в ВС разновидностям модуляции по сочетанию основных параметров – помехоустойчивости и скорости передачи (подп. 2.6.6).

Спектральные составляющие PSK-сигнала с $\delta_{\min}$, равным 0,3 (см. выражение (2.2)), находятся в частотном диапазоне, нижняя и верхняя граничная частота которого могут быть оценены по выражениям (2.4) [1]. В соответствии с выражениями (2.4) и (2.13), максимальная скорость передачи данных методом PSK (DPSK) по ФКС с шириной полосы пропускания Δf, в общем случае, равна примерно $(\Delta f/2)\times\log_2 N$ бит в секунду.

Применение DBPSK и DQPSK предусматривается, например, протоколом модуляции V.22, ориентированным, как и V.21 (см. выше), на представление двоичных данных в АЛС КТСОП. Также использование DBPSK и DQPSK оговаривается рядом стандартов группы IEEE 802.11 (Wi-Fi).

В частности, протокол V.22 оговаривает следующие характеристики DPSK-сигналов [3, 7]:

- частота несущей вызывающего абонента – 1200 Гц;
- частота несущей отвечающего абонента – 2400 Гц;
- скорость передачи данных – 600 бит/с при использовании DBPSK и 1200 бит/с – DQPSK;
- ширина спектра модулированного сигнала – примерно 1200 Гц.

2.6.4. Квадратурная амплитудная модуляция, КАМ (QAM)

Данный тип модуляции представляет собой сочетание амплитудной и фазовой манипуляций, т. е. характеризуется изменением как

амплитуды, так и начальной фазы несущей в процессе кодирования передаваемой цифровой последовательности [1, 5]. Благодаря этому возрастает число «степеней свободы» (различаемых состояний) модулированного сигнала, что, в свою очередь, позволяет представлять в одном такте (периоде модуляции) состояние большего числа бит (до 8-и – 9-и [1, 5]), чем при FSK (1 бит) и PSK (1 – 3 бита). QAM-сигнал формируется как сумма двух амплитудно-манипулированных сигналов, взаимно смещенных по фазе на 90° (т. е. на $\pi/2$), откуда и происходит термин «квадратурная амплитудная модуляция». Математическая модель QAM-сигнала описывается следующим выражением:

$$\begin{aligned} X_{QAM}(t) &= X_{mi} \times \sin\left(\omega_0 t + \varphi_0 + \frac{\pi}{2} + \varphi_{xi}\right) + Y_{mi} \times \sin\left(\omega_0 t + \varphi_0 + \varphi_{yi}\right) = \\ &= X_{mi} \times \cos\left(\omega_0 t + \varphi_0 + \varphi_{xi}\right) + Y_{mi} \times \sin\left(\omega_0 t + \varphi_0 + \varphi_{yi}\right) = \\ &= X_{m0} \times \left\{X_i \times \cos\left(\omega_0 t + \varphi_0\right) + Y_i \times \sin\left(\omega_0 t + \varphi_0\right)\right\} \end{aligned}, \quad (2.21)$$

где X_{mi}, Y_{mi}, φ_{xi} и φ_{yi} – соответственно амплитуды и фазовые сдвиги (относительно несущей) косинусной и синусной компонент QAM-сигнала, представляющего i-ю битовую комбинацию, причем фазовые сдвиги могут принимать только значения 0° и 180° (0 и π);

ω_0 и φ_0 – соответственно угловая частота и начальная фаза несущей;

X_i и Y_i – целочисленные коэффициенты (которые могут быть как положительными, так и отрицательными), соответствующие i-ой битовой комбинации.

Преобразуя выражение (2.21) в соответствии с общеизвестными тригонометрическими равенствами, получаем:

$$\begin{aligned} X_{QAM}(t) &= \sqrt{X_i^2 + Y_i^2} \times X_{m0} \times \sin\left(\omega_0 t + \varphi_0 + arctg\frac{X_i}{Y_i}\right) = \\ &= Am_i \times \sin\left(\omega_0 t + \varphi_0 + \varphi_i\right). \end{aligned} \quad (2.22)$$

Данное выражение наглядно показывает, что QAM-сигнал является модулированным как по амплитуде, так и по фазе.

Удобным способом представления QAM-сигналов является их изображение в виде векторов в сигнальном пространстве, в качестве которого выступает ортогональная система из 2-х координат, служа-

щих для представления соответственно коэффициентов X_i и Y_i математической модели (2.21). Указанный способ иллюстрирует рис. 2.27.

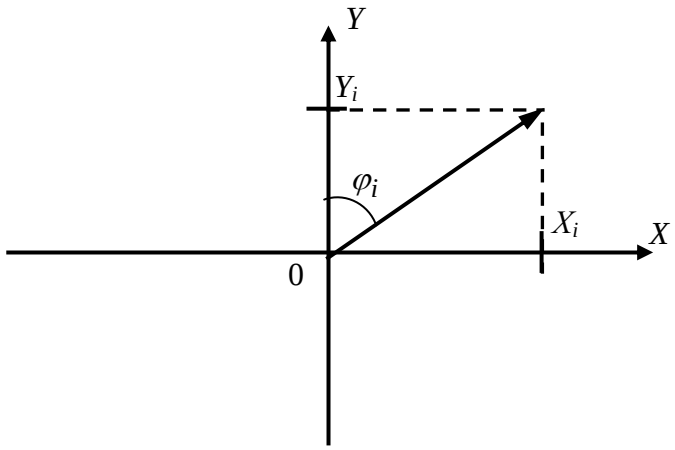


Рис. 2.27. Пример векторного представления QAM-сигнала

Сопоставляя рис. 2.27 с выражением (2.22), нетрудно заметить, что длина изображенного на рис. 2.27 вектора равна амплитуде QAM-сигнала, нормированной относительно амплитуды несущей (т. е. отношению амплитуды QAM-сигнала к амплитуде несущей), а угол между данным вектором и осью ординат – фазовому сдвигу QAM-сиг-нала относительно несущей.

Пример сигнальной диаграммы QAM представлен на рис. 2.28. Данный пример соответствует 16-точечной QAM, оговариваемой одним из профилей протокола модуляции V.32 [1], при котором каждое из сочетаний амплитуды и фазы QAM-сигнала служит для представления определенной 4-битовой комбинации модулирующей цифровой последовательности. Указанные комбинации приведены на рис. 2.28 рядом с соответствующими им точками сигнальной диаграммы. Например, двоичная комбинация *1010* отображается сигналом, описываемым выражением $X_{m0} \times \{-3 \times \cos(\omega_0 t + \varphi_0) + 1 \times \sin(\omega_0 t + \varphi_0)\}$ (см. рис. 2.28), а комбинация *1111* – сигналом, описываемым выражением $X_{m0} \times \{3 \times \cos(\omega_0 t + \varphi_0) + 3 \times \sin(\omega_0 t + \varphi_0)\}$.

Следует отметить, что, кроме сигнальной диаграммы QAM с прямоугольной сеткой сигнальных точек (рис. 2.28), известны также звездообразная и круговая сигнальные диаграммы QAM [1].

Благодаря тому, что QAM-сигнал является модулированным как по амплитуде, так и по фазе, его состояния, представляющие каждую

из возможных кодовых комбинаций, в большей степени различаются между собой, чем у сигнала с таким же количеством различимых состояний, модулированного только по амплитуде или только по фазе. Следовательно, при одинаковом числе различимых состояний, помехоустойчивость КАМ выше, чем при амплитудной или фазовой манипуляции.

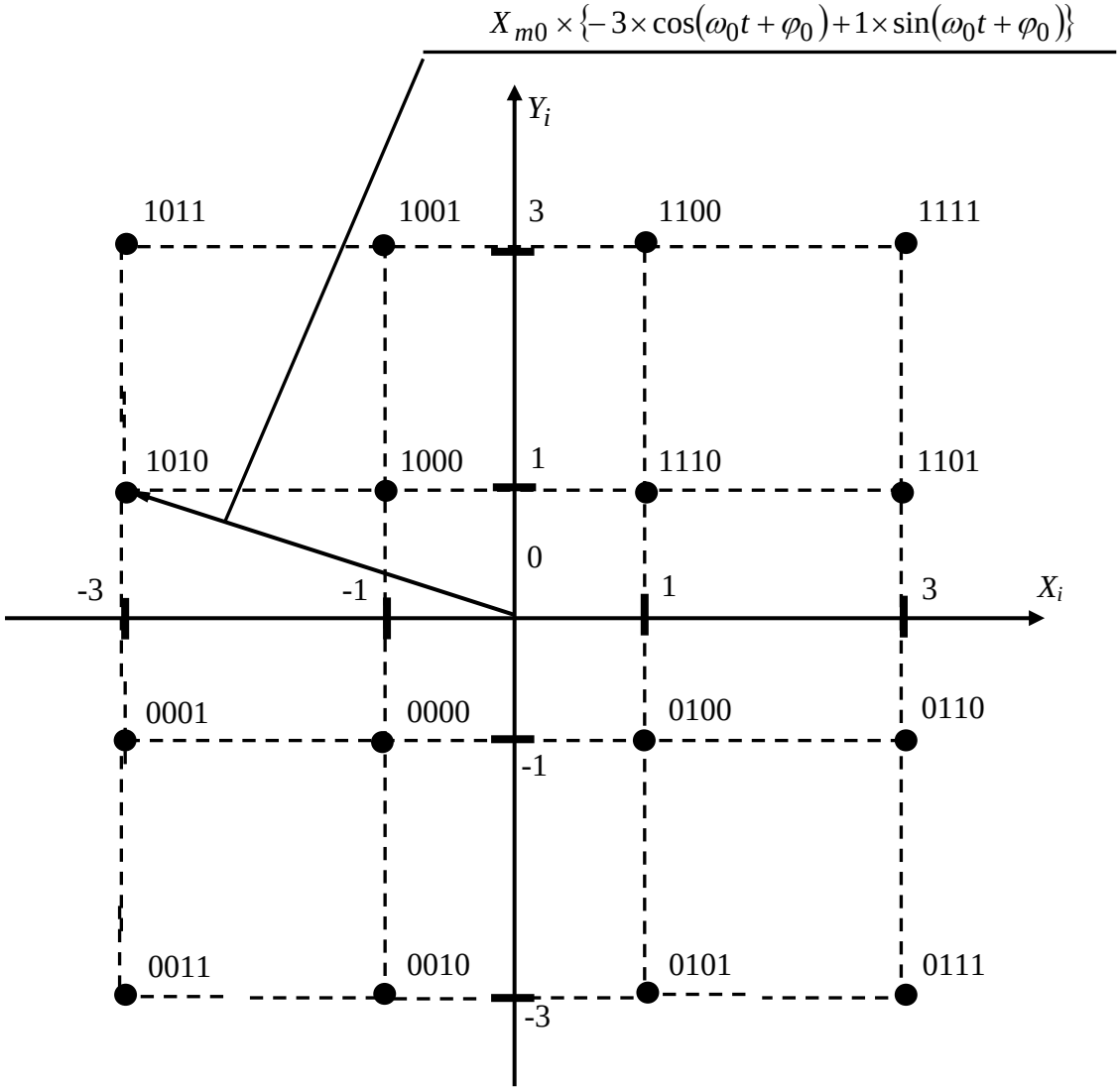
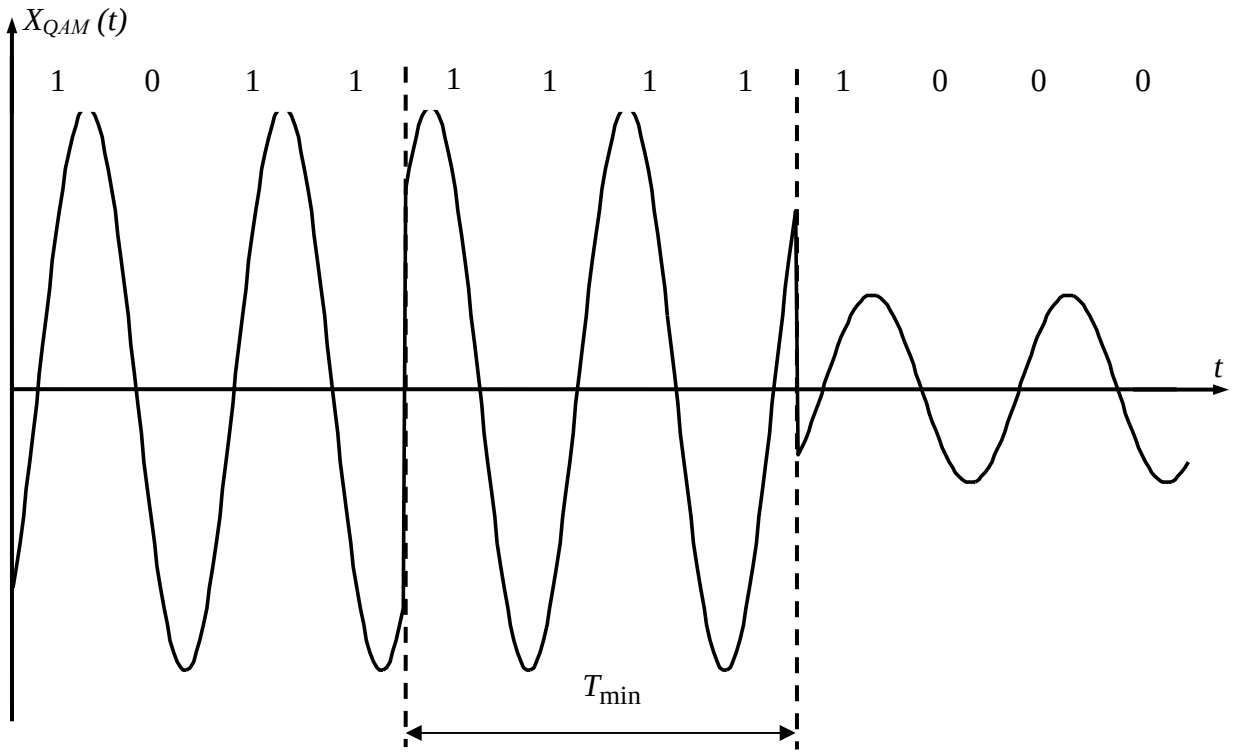


Рис. 2.28. Пример сигнальной диаграммы QAM-16

На рис. 2.29 приведен пример временной диаграммы QAM-сигнала, представляющего двоичную последовательность, приведенную там же. Модуляция соответствует сигнальной диаграмме, приведенной на рис. 2.28.

Следует также отметить, что в настоящее время достаточно широко распространена КАМ *с подавлением несущей*, часто обозначаемая в литературе аббревиатурой *CAP* (от англ. словосочетания *Carrier less Amplitude/Phase modulation*, в дословном переводе – «амп-

литудно-фазовая модуляция без несущей») [1, 5]. Идея CAP-модуляции основывается на следующем. Спектральная компонента амплитудно-модулированного сигнала (в т. ч. QAM-сигнала), частота которой равна частоте несущей, является неизбежным результатом процедуры модуляции, однако не несет информации о модулирующем сигнале.



$T_{\min}$ – минимальная длительность интервала манипуляции
(интервала времени между изменениями амплитуды и фазы QAM-сигнала)

Рис. 2.29. Пример временной диаграммы QAM-сигнала

Поэтому восстановление на приемной стороне исходных данных из QAM-сигнала возможно и при отсутствии в нем указанной спектральной компоненты (при условии, что частота несущей «известна» приемнику). С другой стороны, наличие несущей в амплитудно-модулированном сигнале предъявляет повышенные требования к мощности передающей аппаратуры, а также увеличивает уровень перекрестных помех при частотном уплотнении каналов связи, т. е. при передаче нескольких модулированных сигналов с различными частотами несущей по одному и тому же ФКС. Поэтому перед передачей амплитудно-модулированных сигналов (в т. ч. QAM-сигналов) по линиям связи, вообще говоря, рационально подавлять их несущую, с ее восстановлением специальными методами на приемной стороне.

Основным недостатком такого подхода по сравнению с амплитудной модуляцией без подавления несущей является относительная сложность его реализации, которая до сравнительно недавнего времени препятствовала его широкому распространению. Однако в настоящее время, благодаря наличию эффективных аппаратно-программных средств цифровой обработки сигналов, этот недостаток не является существенным. Указанное обусловливает применение CAP, например, в беспроводных ФКС, а также в технологиях DSL, для которых характерно частотное уплотнение каналов связи (см. п. 2.8.2).

В общем случае, граничные частоты спектра QAM-сигнала при $\delta_{\min}$, равном 0,3 (см. (2.2)), оцениваются по выражениям (2.4). Однако на практике, за счет применения специальных мер, в том числе предварительного логического кодирования модулирующей двоичной последовательности (п. 2.7) частотный диапазон QAM-сигнала может быть сужен до граничных частот, примерно равных $f_0 - (0{,}5/T_M)$ и $f_0 + (0{,}5/T_M)$ соответственно [1, 5]. Ширина частотного диапазона QAM-сигнала при этом составляет примерно $1/T_{\min}$, а максимальная скорость передачи данных при ширине полосы пропускания ФКС, равной Δf, составляет, в соответствии с выражением (2.13), $\Delta f \times \log_2 N$ бит/с, где N – число точек сигнальной диаграммы КАМ. Разрядность битовой комбинации (*символа*), представляемой в каждом интервале КАМ, равна $\log_2 N$.

КАМ, при прочих равных условиях, обеспечивает наибольшую скорость передачи информации (в битах в секунду) из всех методов модуляции с синусоидальной несущей [1]. Необходимо при этом отметить, что скоростные протоколы КАМ предусматривают предварительное логическое кодирование модулирующей двоичной последовательности (п. 2.7), которое позволяет частично устранить ошибки обмена данными, вызываемые помехами в канале связи и его ограниченной полосой пропускания. Данный прием, в свою очередь, обеспечивает повышение скорости передачи при заданной ширине полосы пропускания канала [1, 5].

КАМ лежит в основе многих протоколов модуляции, ориентированных на применение как в АЛС КТСОП [7], в частности, в ряде технологий группы xDSL (см. подп. 2.2.1), так и в беспроводных ФКС, например, в мобильной связи и в беспроводных WiFi-ЛВС [5]. Типовым примером использования КАМ являются протоколы

группы V.34, ранее широко использовавшиеся для представления двоичных данных в АЛС КТСОП [7]. Базовый протокол этой группы оговаривает следующие характеристики QAM-сигнала:

- частота несущей не является фиксированной (с целью максимально эффективного использования частотного диапазона абонентского канала КТСОП) и выбирается из ряда 1600, 1646, 1680, 1800, 1829, 1867, 1920, 1959 или 2000 Гц;
- частоты манипуляции, определяемые, как $1/T_{\min}$ (см. рис. 2.29) – 2400, 3000 или 3200 Гц (опционально – также 2743, 2800 и 3429 Гц);
- базовые скорости передачи данных (при частоте манипуляции 2400 Гц) – от 2400 до 28800 бит/с (в более поздних протоколах данной группы – до 33600 бит/с), с шагом 2400 бит/с;
- число точек сигнального созвездия – до 960-ти (например, 24, 120, 240 и 960 при скоростях передачи 9600, 19 200, 24 000 и 28 800 бит/с соответственно);
- метод логического кодирования модулирующей цифровой последовательности – самосинхронизирующееся скремблирование с последующим решетчатым кодированием (п. 2.7).

Интересными особенностями протоколов группы V.34 являются [7]:

- возможность организации асимметричных каналов связи между вызывающим и отвечающим абонентами, т. е. передачи данных в противоположных направлениях с разными скоростями и/или с различающимися между собой частотами несущих;
- возможность использования для обмена служебной информацией между абонентами дополнительного канала связи в каждом из направлений, со скоростью передачи данных 200 бит/с; указанный канал организуется методом *временного мультиплексирования* (п. 2.8), т. е. под него выделяются специальные интервалы времени в сеансе связи.

При этом, в отличие от низкоскоростных протоколов V.21 и V.22 (см. подп. 2.6.2 и 2.6.3), сигналы вызывающего и отвечающего абонентов (т. е. входящей и исходящей связи) не разделены по частоте, и под каждый из них выделен весь частотный диапазон АЛС КТСОП (от 300 до 3400 Гц). Это позволяет обеспечить скорость обмена данными вдвое большую (при прочих равных условиях), чем при выделении под каждый из указанных сигналов половины указанного диапазона, как в протоколах V.21 и V.22. Однако выделение одного и того же частотного диапазона под сигналы как входящей, так и исходящей связи требует применения специальных средств эхоподавления для предотвращения взаимных искажений указанных сигналов.

Другим показательным примером КАМ является *дискретная мультитональная КАМ*, обычно обозначаемая в литературе аббревиатурой DMT (от англ. словосочетания Discrete Multi-Tone). Она применяется, например, для обмена данными между абонентским ПК и провайдером в технологии ADSL (см. подп. 2.2.1). Основным отличием DMT от «классических» протоколов КАМ, например, V.34 (см. выше), является применение не одной, а множества несущих, каждая из которых подвергается КАМ независимо от других и используется для передачи определенной части битового трафика между абонентским ПК и провайдером.

В соответствии с одним из базовых протоколов реализации ADSL-DMT [7], диапазон частот от 26 кГц до 1,1 МГц, отведенный для связи между абонентским ПК и провайдером, разделяется на 249 поддиапазонов (*частотных каналов*) шириной 4312,5 Гц каждый. Из них по одному отводится для контроля передачи данных от провайдера к абоненту и в противоположном направлении, 224 – для передачи пользовательских данных от провайдера к абоненту, остальные – для пользовательского трафика от абонента к провайдеру. На практике обычно бывают задействованы не все частотные каналы, выделенные для обмена пользовательскими данными между абонентом и провайдером, а только те из них, уровень помех и степень затухания сигнала в частотном диапазоне которых позволяют обеспечить приемлемое качество связи. При этом определение указанных характеристик частотных каналов, как и назначение каналов, реально отводимых для передачи данных, осуществляются в процессе связи автоматически.

В пределах каждого из задействованных частотных каналов передача цифровых данных осуществляется методом КАМ в соответствии с протоколом, в общем, аналогичным протоколу V.34 (см. подп. 2.3.4), но, в отличие от него, со скоростью 4000 (а не 2400) символов в секунду. Частота несущей каждого из каналов при этом выбирается равной примерно середине частотного диапазона, выделенного под соответствующий канал. Разрядность символа может быть от 2-х до 15-ти бит. Она устанавливается автоматически в процессе связи независимо для каждого из частотных каналов, как максимальное число достоверных бит, которое может быть передано за один период модуляции при текущем уровне помех и степени затухания сигнала в диапазоне частот соответствующего канала.

Абонентский ADSL-модем представляет собой, по существу, цифровой сигнальный процессор, выполняющий функции 249-и QAM-модемов. Аналогичное устройство входит в состав блока доступа к цифровой абонентской линии (см. рис. 2.7).

Нетрудно увидеть, что, если задействованы все 224 частотных канала, выделенных для входящего трафика абонента, а разрядность символа равна 15 бит, скорость передачи данных от провайдера к абоненту составит 224×15×4000 = 13 440 000 бит/с. Данное значение является теоретически максимальным для вышеописанного протокола обмена данными между абонентом ADSL и провайдером. При этом практически достижимый максимум скорости входящего трафика абонента – порядка 8 Мбит/с, а большинство реальных АЛС КТСОП обеспечивает значение скорости передачи данных от провайдера к абоненту порядка 1 – 2 Мбит/с при скорости исходящего трафика абонента от 64 до 256 Кбит/с [7].

Ряд технологий группы xDSL использует также КАМ с подавлением несущей, CAP (см. выше).

2.6.5. Импульсно-кодовая модуляция (ИКМ)

Данный метод модуляции обозначается также аббревиатурой *PCM* (от англ. словосочетания *Pulse Code Modulation*). Необходимо отметить, что, вообще говоря, термин «ИКМ» применяется к процедуре передачи аналоговых данных по цифровым линиям связи [3]. Однако при передаче цифровых данных по аналоговой линии связи указанный термин используется для *многоуровневой амплитудно-импульсной модуляции* (*Pulse Amplitude Modulation*, *PAM*) [3, 7]. ИКМ (PAM) отличается от ранее рассмотренных методов модуляции тем, что в качестве несущей при ней выступает не синусоидальный сигнал, а *последовательность прямоугольных импульсов*.

Принцип ИКМ (PAM) применительно к передаче цифровых данных по аналоговым линиям связи иллюстрирует рис. 2.30.

Подлежащие передаче данные представляются в виде последовательности k-разрядных двоичных чисел. Она, посредством ЦАП такой же разрядности, преобразуется в ступенчато изменяющийся аналоговый сигнал, представляющий собой последовательность примыкающих друг к другу прямоугольных импульсов с амплитудой, прямо пропорциональной указанным числам, т. е. в PAM-сигнал, показан-

ный на рис. 2.30 более тонкой линией. PAM-сигнал затем подвергается сглаживанию посредством включенного после ЦАП фильтра нижних частот; сглаженный сигнал обозначен на рис. 2.30 более интенсивной линией.

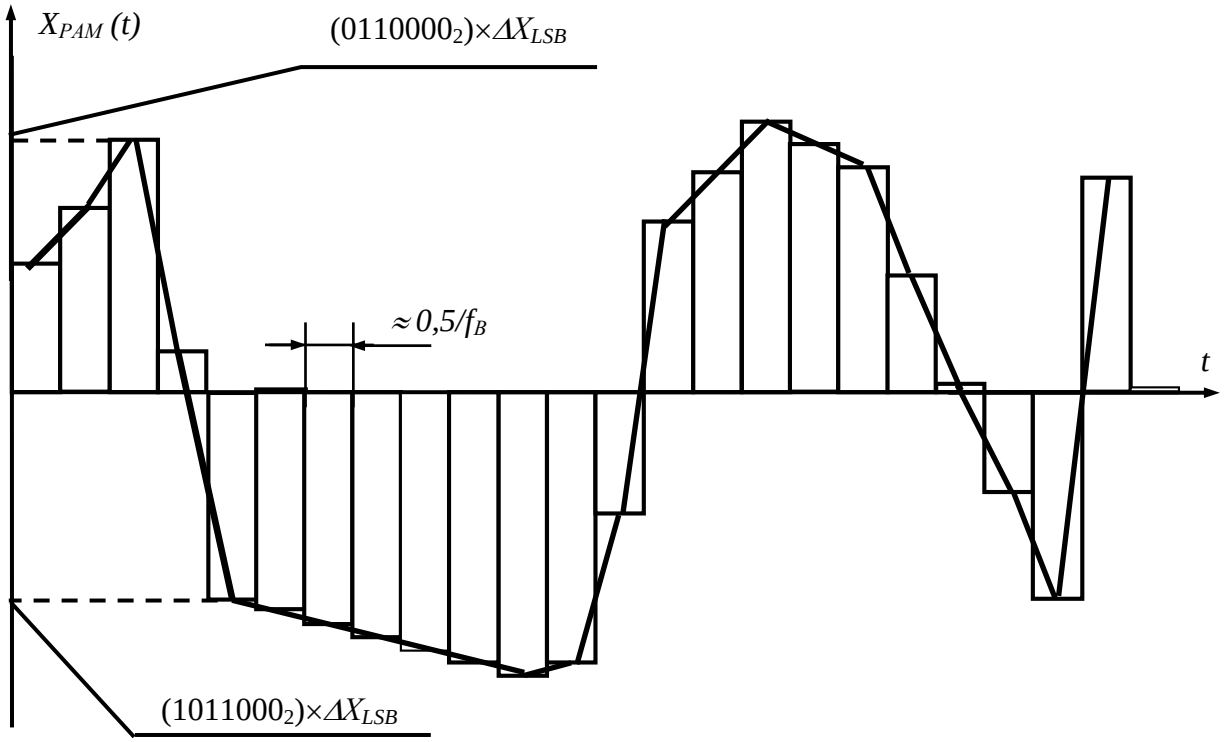


Рис. 2.30. Пример временной диаграммы PAM-сигнала до и после сглаживания (более тонкая и более интенсивная линии соответственно): $xxxxxxx_2$ – двоичные числа в дополнительном коде, соответствующие передаваемым цифровым данным; f_B – верхняя граничная частота полосы пропускания канала связи; ΔX_{LSB} – квант (минимально возможное изменение) выходного сигнала ЦАП

При этом частота поступления модулирующих двоичных чисел на ЦАП выбирается на 10 – 20 % большей, чем удвоенная верхняя граничная частота полосы пропускания канала связи, а представляемая PAM-сигналом двоичная последовательность подвергается предварительному логическому кодированию (п. 2.7) и предыскажению специальным цифровым фильтром. В результате, спектр PAM-сигнала после сглаживания (более интенсивная линия на рис. 2.30) укладывается в полосу пропускания ФКС [3, 7].

На приемной стороне из модулированного аналогового сигнала посредством АЦП восстанавливается исходная последовательность цифровых данных. Естественно, однозначное восстановление исход-

ных цифровых данных из РАМ-сигнала возможно только при условии, что уровень шумов и помех на линии связи не превышает значения одного кванта выходного сигнала ЦАП (т. е. аналогового эквивалента его младшего значащего разряда). Поэтому ИКМ (РАМ), в целом, предъявляет более жесткие требования к отношению сигнал-шум на линии, чем КАМ, и, тем более, чем ДОФМ или ЧМ (подп. 2.6.6).

На практике вышеописанный метод ИКМ (РАМ) ранее широко применялся для обмена данными между Интернет-провайдером и абонентом КТСОП (см. рис. 2.6), при следующих условиях:

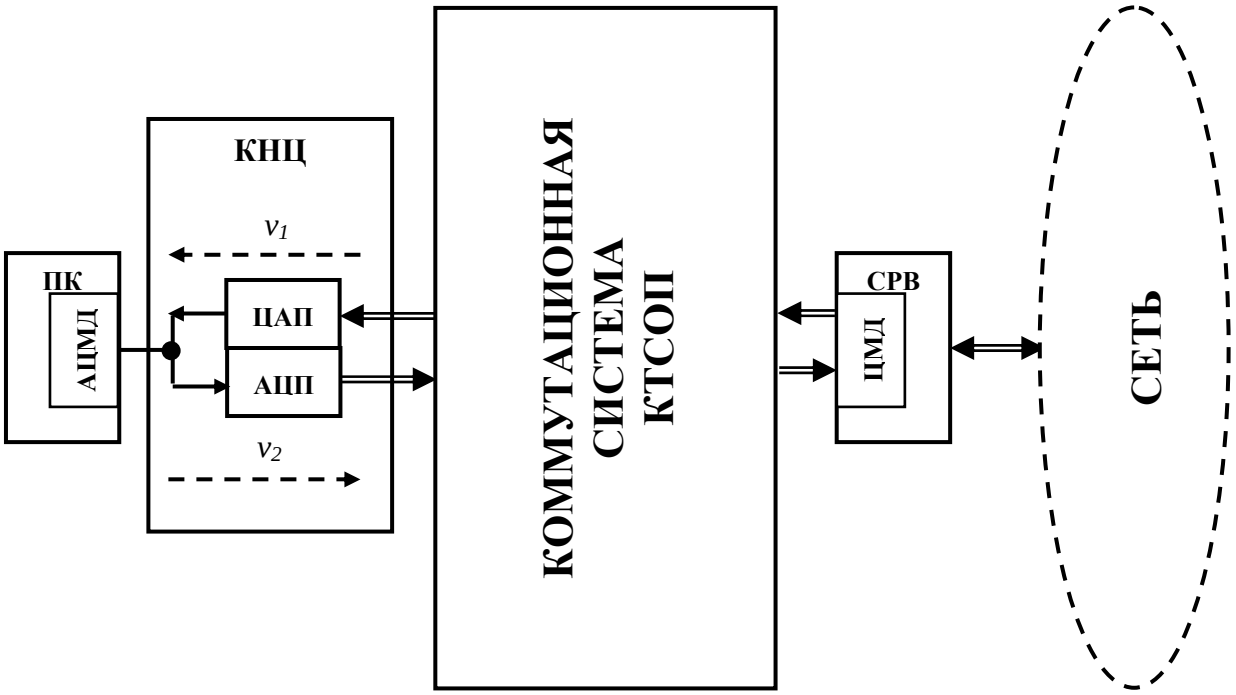
- представлении данных в аналоговом виде только в АЛС КТСОП и в цифровом – на всех остальных ее участках, что имеет место практически для всех современных КТСОП [3] (см. подп. 2.2.1);
- цифровом представлении данных в ФКС между провайдером и КТСОП и отсутствии организационно-законодательных ограничений на его полосу пропускания.

Принцип обмена данными между провайдером и абонентом по КТСОП на основе метода ИКМ (РАМ) поясняет рис. 2.31.

Цифровые данные от провайдера к абоненту передаются через КТСОП в соответствии со стандартом их представления в ней. Наиболее распространена на практике передача указанных данных через КТСОП в виде совокупности 7- или 8-битовых двоичных слов, частота следования которых равна 8 кГц [3]. В абонентском концентраторе данная последовательность двоичных слов посредством ЦАП преобразуется в аналоговый сигнал, частотный диапазон которого находится в пределах граничных частот полосы пропускания абонентского ФКС КТСОП (от 300 до 3400 Гц). Указанный сигнал поступает на абонентский модем по АЛС, соединяющей абонента с концентратором. Входящий в состав аналого-цифрового абонентского модема АЦП преобразует указанный сигнал в исходную последовательность двоичных слов, представляющих собой данные от провайдера.

В свою очередь, передача данных от абонента к провайдеру может осуществляться или методом ИКМ (РАМ), или посредством модулированного сигнала с синусоидальной несущей [3]. Необходимо отметить, что на практике объем данных, поступающих от провайдера к абоненту, обычно в несколько раз превышает аналогичный параметр потока информации в противоположном направлении [3, 7]. Следовательно, требования к скорости передачи данных от провайдера к абоненту, вообще говоря, более высокие, чем к скорости передачи от абонента к провайдеру. Поэтому последняя часто осуществляется на

основе менее скоростного способа модуляции, чем передача данных от провайдера к абоненту КТСОП. Такой подход применяется в ряде протоколов модуляции, использующих ИКМ (PAM) (см. далее).



– цифровые линии связи;
– аналоговые линии связи;
– потоки данных;

ПК – абонентский персональный компьютер; АЦМД – аналого-цифровой модем;
КНЦ – концентратор КТСОП; СРВ – сервер провайдера;
ЦМД – цифровой модем; v_1, v_2 – скорости передачи данных соответственно от провайдера к абоненту и в противоположном направлении

Рис. 2.31. Принцип обмена данными с применением ИКМ (PAM) между Интернет-провайдером и абонентом КТСОП

Вышеописанный принцип обмена данными позволяет обеспечить скорость передачи данных, потенциально большую, чем могут обеспечить методы модуляции с синусоидальной несущей [3, 7]. Например, при представлении данных в КТСОП 8-битовыми словами с частотой следования 8 кГц значение данной скорости теоретически может достигать 64 000 бит/с, реально – до 56 000 бит/с [1].

Применение метода ИКМ (PAM) предусматривается, в частности, протоколами V.90 и V.92 [7]. Оба указанных протокола ориентированы на представление цифровых данных в АЛС КТСОП. Протокол V.90 оговаривает применение метода ИКМ (PAM) для передачи данных от провайдера к абоненту со скоростью от 28 000 до 56 000 бит/с.

Для передачи информации в обратном направлении используется КАМ в соответствии с базовым протоколом группы V.34. Протокол V.92 предусматривает применение ИКМ (PAM) для передачи данных в обоих направлениях, от абонента к провайдеру – со скоростью от 24 000 до 48 000 бит/с, а в противоположном направлении – от 28 000 до 56 000 бит/с.

Частота передачи символов методом ИКМ (PAM) в соответствии как с протоколом V.90, так и V.92 одинакова для всех скоростей передачи и составляет 8000 Гц (8000 символов/с), т. е. примерно на 18 % выше удвоенной верхней граничной частоты полосы пропускания абонентского канала КТСОП, равной 2×3400 Гц [3, 7]. Различные скорости передачи различаются между собой разрядностью символа, устанавливаемой автоматически в процессе обмена данными, и определяемой числом достоверных бит результата аналого-цифрового преобразования PAM-сигнала (см. рис. 2.30) при реально существующем в канале связи уровне помех.

Следует отметить, что скорость обмена данными 56 000 бит/с является практически предельной при соединении абонента ВС с провайдером по абонентскому ФКС КТСОП с полосой пропускания от 300 до 3400 Гц [3, 7].

2.6.6. Техническая реализация модуляции/демодуляции

Данная реализация осуществляется посредством модемов абонентских компьютеров и функционально аналогичных блоков коммутационных устройств ВС (коммутаторов, маршрутизаторов и т. п.), а также программного обеспечения этого оборудования. Формирование модулированных сигналов, как правило, осуществляется в программной форме, с последующим аппаратным преобразованием описывающей эти сигналы кодовой последовательности в напряжение, ток или уровень излучения. Извлечение из модулированных сигналов представляемых ими данных на приемной стороне (демодуляция) обычно реализуется также в программной форме (в частности, методами быстрого преобразования Фурье в течение каждого из интервалов модуляции), после предварительного аналого-цифрового преобразования модулированных сигналов и программной цифровой фильтрации его результатов.

Модемы и аналогичные им блоки коммутационных устройств ВС, кроме функций физического уровня модели OSI, реализуют также и функции канального уровня. Поэтому принципы их реализации будут рассмотрены далее, в гл. 3.

Выводы по п. 2.6

В табл. 2.6 представлены основные характеристики способов модуляции, рассмотренных в подп. 2.6.2 – 2.6.5 [1, 3, 5].

Таблица 2.6

Базовые характеристики распространенных разновидностей методов модуляции, рассмотренных в подп. 2.6.2 – 2.6.5

Способ модуляции	Разрядность символа, представляемого в одном интервале манипуляции, бит	Максимальная скорость обмена данными (в общем случае, без применения предварительного логического кодирования), бит/с*	Максимально допустимое отношение «сигнал-шум» в ФКС (по мощности), дБ**
FSK			
Двухчастотная узкополосная FSK	1	$\Delta f / 2$	10,5
PSK			
DBPSK	1	$\Delta f / 2$	11,2
DQPSK	2	Δf	14,5
DOPSK***	3	$1{,}5 \times \Delta f$	16,8
QAM			
QAM-8	3	$1{,}5 \times \Delta f$	18,0
QAM-16	4	$2 \times \Delta f$	21,5
QAM-32	5	$2{,}5 \times \Delta f$	24,5
QAM-64	6	$3 \times \Delta f$	27,7
QAM-128	7	$3{,}5 \times \Delta f$	30,6
QAM-256	8	$4 \times \Delta f$	33,8
PAM			
PAM-32	5	$10 \times \Delta f$	30,1
PAM-64	6	$12 \times \Delta f$	36,1
PAM-128	7	$14 \times \Delta f$	42,1
PAM-256	8	$16 \times \Delta f$	48,2

* Δf – ширина полосы пропускания ФКС;

**для обеспечения BER$\leq 10^{-7}$ (см. подп. 2.2.3);

***DOPSK (8-точечная DPSK) применяется достаточно редко; ее характеристики представлены в настоящей таблице с целью обоснования относительно малой распространенности на практике DPSK с числом точек сигнальной диаграммы, большим 4-х

Из табл. 2.6 нетрудно заметить, что чем большую скорость обмена данными обеспечивает тот или иной способ модуляции, тем меньшей помехоустойчивостью он обладает (т. е. тем большего отношения «сигнал-шум» в ФКС он требует). Так как уровень шумов в ФКС обычно изменяется во времени случайным образом и в достаточно широких пределах, на практике обычно весьма затруднительно или невозможно выбрать какой-либо конкретный способ модуляции, наиболее приемлемый для некоторого конкретного ФКС. Поэтому подавляющее большинство современных модемов являются *мультистандартными*, т. е. обладающими возможностью реализации различных методов модуляции, а также различных вариантов реализации каждого из методов. В процессе работы модем в зависимости от текущего уровня шумов в ФКС автоматически выбирает способ модуляции, обеспечивающий максимальную скорость устойчивого обмена данными при текущем отношении «сигнал-шум» в ФКС.

Основными областями применения рассмотренных в подп. 2.6.2 – 2.6.5 методов модуляции (манипуляции) являются следующие.

Частотная манипуляция (FSK), в том числе такие ее разновидности, как MSK и GMSK (см. подп. 2.6.2), благодаря высокой помехоустойчивости, применяются в основном при значительном уровне шумов и помех в ФКС. Это имеет место, например, в ряде АЛС КТСОП (как правило, не постоянно, а в течение некоторых случайных интервалов времени), а также в ряде практических случаев в беспроводных ФКС. Поэтому применение узкополосной FSK предусматривается, например, протоколом V.21 [1], ранее использовавшимся в зашумленных АЛС КТСОП (преимущественно в качестве «аварийного» и на этапе установления соединения), а GMSK – одним из стандартов группы IEEE 802.11 (Wi-Fi) и рядом стандартов группы GSM [5].

Из существующих разновидностей *фазовой манипуляции* (PSK) на практике наиболее распространена дифференциальная PSK (DPSK), благодаря устойчивости к нестабильности параметров детектора и фазы модулированного сигнала (см. подп. 2.6.3). При этом практический смысл имеют в основном двух- и четырехточечная DPSK, DBPSK и DQPSK соответственно (см. рис. 2.25, *б* и 2.26). DPSK с числом точек сигнальной диаграммы (т. е. с числом различаемых значений фазового сдвига) от 8-ми и более не нашла широкого применения, так как уступает QAM с аналогичным количеством точек по сочетанию помехоустойчивости и скорости передачи данных (см. табл. 2.6). По-

мехоустойчивость DBPSK и DQPSK сопоставима с таковой двухчастотной FSK (см. табл. 2.6). Обеспечиваемая ими скорость обмена примерно равна (у DBPSK) или в 2 раза больше (у DQPSK) скорости, обеспечиваемой FSK при такой же ширине полосы пропускания ФКС. Благодаря названным свойствам DBPSK и DQPSK, они, как и FSK, применяются в протоколах модуляции, ориентированных на использование в зашумленных АЛС КТСОП (например, V.22 [1]), а также в беспроводных ФКС. В частности, применение DBPSK и DQPSK оговаривается рядом стандартов группы IEEE 802.11 (Wi-Fi) [5], а также стандартов группы IEEE 802.16, регламентирующих вопросы реализации беспроводных технологий связи WiMAX [5].

Квадратурная амплитудная модуляция (QAM), в том числе QAM с подавлением несущей (CAP) достаточно широко применяется для представления двоичных данных как в беспроводных ФКС, так и в абонентских ФКС КТСОП. Это обусловлено большим разнообразием имеющих практический смысл вариантов реализации, отличающихся между собой числом точек сигнальной диаграммы и, как следствие, соотношениями скорости обмена данными и помехоустойчивости (см. табл. 2.6). Поэтому существует широкая возможность выбора варианта QAM, наиболее приемлемого по сочетанию названных характеристик для каждой из большинства реальных ФКС, требующих применения модулированных сигналов-носителей данных (см. п. 2.4). Исключение составляют сильно зашумленные ФКС (требующие применения двухчастотной FSK, DBPSK или DQPSK) или, наоборот, ФКС с уровнем шумов, достаточно низким для возможности применения PAM (см. табл. 2.6). В частности, применение различных вариантов реализации QAM оговаривается:

- рядом протоколов модуляции, ориентированных на применение в АЛС КТСОП, например, V.32 и V.34 [1, 7];
- стандартами ADSL [7];
- рядом стандартов группы IEEE 802.16 (WiMAX) [5];
- рядом стандартов мобильной связи [5].

Импульсно-кодовая модуляция (PAM), как видно из табл. 2.6, является наиболее скоростным из рассмотренных в подп. 2.6.2 – 2.6.5 методов модуляции. Например, при заданной ширине полосы пропускания ФКС, PAM обеспечивает в 2 раза большую скорость обмена, чем QAM с таким же количеством различаемых состояний сигнала. Однако PAM отличается наименьшей помехоустойчивостью (см. табл. 2.6). Поэтому применение PAM ограничивается ФКС, тре-

бующими использования модулированных сигналов-носителей и в то же время отличающимися низким уровнем шумов, в основном – высококачественными АЛС КТСОП [7].

В заключение следует отметить, что большинство применяемых в настоящее время протоколов модуляции (особенно КАМ и ИКМ) предполагает использование предварительного *логического кодирования* исходной двоичной последовательности (п. 2.7). Оно позволяет приблизить скорость устойчивого обмена данными к максимально возможной при заданной ширине полосы пропускания ФКС, а также удовлетворить условию (2.14) при любом возможном сочетании нулей и единиц исходной двоичной последовательности.

2.7. Логическое кодирование двоичных данных в ФКС ВС

2.7.1. Общие положения

Как указано в п. 2.4, логическое кодирование применительно к ФКС ВС состоит в предварительном преобразовании подлежащей передаче по ФКС двоичной последовательности в некоторую другую, непосредственно передаваемую по ФКС методом линейного кодирования или модуляции. На приемной стороне после линейного декодирования или, соответственно, демодуляции, из указанной последовательности восстанавливаются исходные двоичные данные (см. рис. 2.18). Основными целями применения логического кодирования являются следующие [1, 5]:

- выполнение условий (2.14) и, при необходимости, (2.15) для любого возможного сочетания нулей и единиц в передаваемой двоичной последовательности;
- повышение устойчивости процесса обмена данными к ошибкам, обусловленным ограниченной полосой пропускания ФКС и помехами в нем.

Кроме того, на предварительном логическом кодировании передаваемых по ФКС двоичных данных основывается ряд методов *мультиплексирования ФКС*, т. е. использования несколькими ФКС одного и того же участка передающей среды [1, 3, 5]. Методы логического кодирования, используемые при мультиплексировании ФКС, будут рассмотрены далее, в посвященном мультиплексированию п. 2.8.

Основными методами логического кодирования, применяемыми для достижения двух вышеперечисленных целей, являются [1, 3, 5]:

- избыточное RLL-кодирование;
- скремблирование;
- сверточное кодирование.

Рассмотрим сущность, принципы реализации и основы применения каждого из этих методов.

2.7.2. Избыточное RLL-кодирование

Название данного метода происходит от англ. словосочетания «Run-Length Limited», в дословном переводе – «ограниченная длина пробега». Смысл этого названия состоит в том, что число следующих подряд нулей или/и единиц в RLL-коде ограничено определенным значением, зависящим от конкретной разновидности кода. Благодаря этому свойству RLL-кодов они применяются преимущественно для обеспечения условия (2.14).

Принцип RLL-кодирования исходной двоичной последовательности состоит в следующем. Указанная последовательность разбивается на фрагменты с некоторой разрядностью m, зависящей от конкретной разновидности избыточного кода. Затем каждый из этих фрагментов заменяется на новый, с разрядностью n, большей m. Замена осуществляется таким образом, чтобы сигнал-носитель, представляющий полученную в результате избыточного кодирования последовательность битов, удовлетворял условию (2.14); у ряда разновидностей RLL-кодов – также условию (2.15).

Типовым примером избыточного RLL-кода является код 4B/5B, применяемый в технологиях ЛВС FDDI и Fast Ethernet [3]. В соответствии с этим кодом, каждые 4 бита исходной двоичной последовательности заменяются 5-битовым фрагментом согласно табл. 2.7. Из данных табл. нетрудно заметить, что полученная в результате 4B/5B-кодирования битовая последовательность не содержит более 2-х идущих подряд нулей. Следовательно, при ее представлении, например, NRZ-1 или MLT-3-кодом, практически полностью удовлетворяется условие (2.14).

Важным дополнительным преимуществом избыточного RLL-кода является возможность *обнаружения ошибок* на линии. В самом деле,

общее количество битовых комбинаций *n*-разрядного фрагмента кода больше, чем соответствующего ему *m*-разрядного фрагмента исходной двоичной последовательности.

Таблица 2.7

Соответствие битовых комбинаций кода 4B/5B 4-битовым комбинациям исходной двоичной последовательности

Исходная 4-битовая комбинация	Код 4B/5B	Исходная 4-битовая комбинация	Код 4B/5B
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

Поэтому из 2^n возможных комбинаций RLL-кода только 2^m являются *разрешенными*, а остальные – *запрещенными* (в англоязычной литературе – *code violations*), поступление которых на вход RLL-декодера принимающего абонента является признаком искажения сигнала-носителя данных при его передаче по ФКС. Например, число возможных 5-битовых комбинаций кода 4B/5B равно 32, в то время как соответствующих им 4-битовых фрагментов исходной последовательности – только 16. Поэтому только 16 5-битовых комбинаций 4B/5B-кода являются разрешенными, т. е. соответствующими какому-либо 4-битовому фрагменту исходной последовательности, а остальные – запрещенными, являющимися признаком искажения сигнала-носителя в ФКС.

Кроме избыточного кода 4B/5B, естественно, существуют другие разновидности RLL-кодов [1, 5].

RLL-кодирование/декодирование обычно осуществляется посредством таблиц перекодировки, реализуемых программно или в аппаратной форме (на основе ПЗУ). Табличное кодирование/декодирование является достаточно простой операцией, практически не усложняющей АПД ФКС.

RLL-кодирование применяется в основном в кабельных ФКС ЛВС, для устранения длинных последовательностей нулей или/и единиц в двоичных последовательностях, подвергаемых линейному

кодированию [1]. В частности, как указано выше, использование RLL-кодирования оговаривается рядом протоколов физического уровня технологий ЛВС FDDI и Fast Ethernet [3].

2.7.3. Скремблирование

Название данного метода логического кодирования происходит от английского глагола «*to scramble*», в дословном переводе – «перемешивать, взбалтывать». Сущность скремблирования [1, 3] состоит в явном или неявном суммировании по модулю 2 исходного сообщения с *псевдослучайной* двоичной последовательностью, обладающей следующими основными свойствами:

- характером чередования нулей и единиц, свойственным случайной последовательности;
- воспроизводимостью и повторяемостью указанного чередования (в отличие от полностью случайной двоичной последовательности);
- совпадением длительностей и начал тактовых интервалов с таковыми скремблируемой двоичной последовательности.

Псевдослучайный характер последовательности, суммируемой с исходным двоичным кодом, придает скремблированной двоичной последовательности следующие свойства [1, 3]:

- гарантированное отсутствие длинных последовательностей нулей и единиц;
- близкое к нулю значение постоянной составляющей при представлении биполярным линейным кодом;
- практическое совпадение граничных частот амплитудного спектра линейных кодов, представляющих двоичную последовательность до и после скремблирования;
- возможность однозначного восстановления исходной двоичной последовательности из скремблированного сигнала, благодаря воспроизводимости и повторяемости псевдослучайного двоичного кода, используемого при скремблировании.

Благодаря вышеперечисленным свойствам, любой из рассмотренных ранее линейных кодов или способов модуляции при представлении скремблированной двоичной последовательности практически полностью удовлетворяет условиям (2.14) и (2.15). Основным недостатком скремблирования по сравнению с избыточным кодированием является относительная сложность реализации (см. далее). Однако

данный недостаток не является существенным при современном уровне развития аппаратно-программных средств АПД. Поэтому скремблирование как логическое кодирование достаточно широко применяется в различных технологиях ВС [1].

Практическая реализация скремблирования осуществляется на основе генераторов псевдослучайных последовательностей (ГПП). Наиболее распространенная операционная модель ГПП представлена на рис. 2.32 [1]. Термин «операционная модель» (а не структурная или функциональная схема) здесь применен потому, что в настоящее время, наряду с аппаратной, широко распространена программная реализация ГПП, базирующаяся на моделировании приведенной на рис. 2.32 структуры в программной форме.

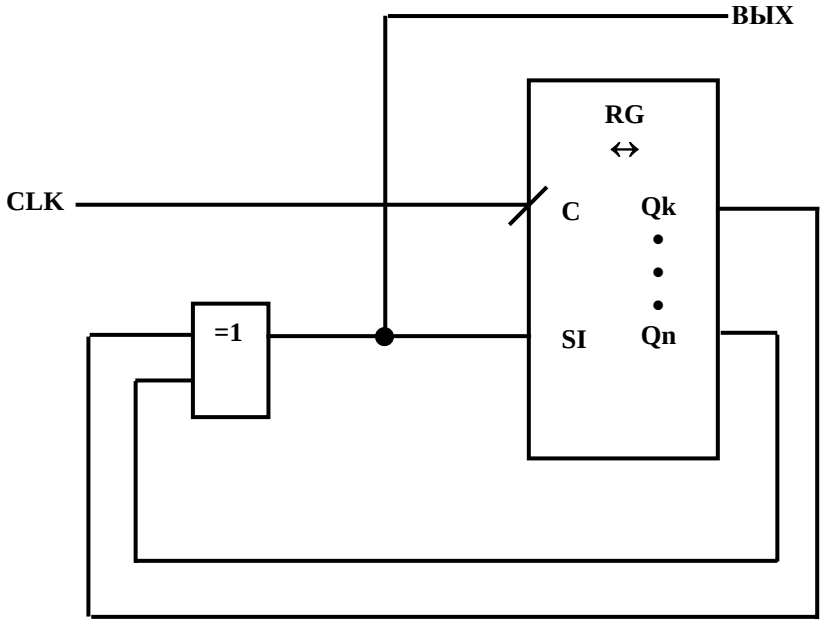


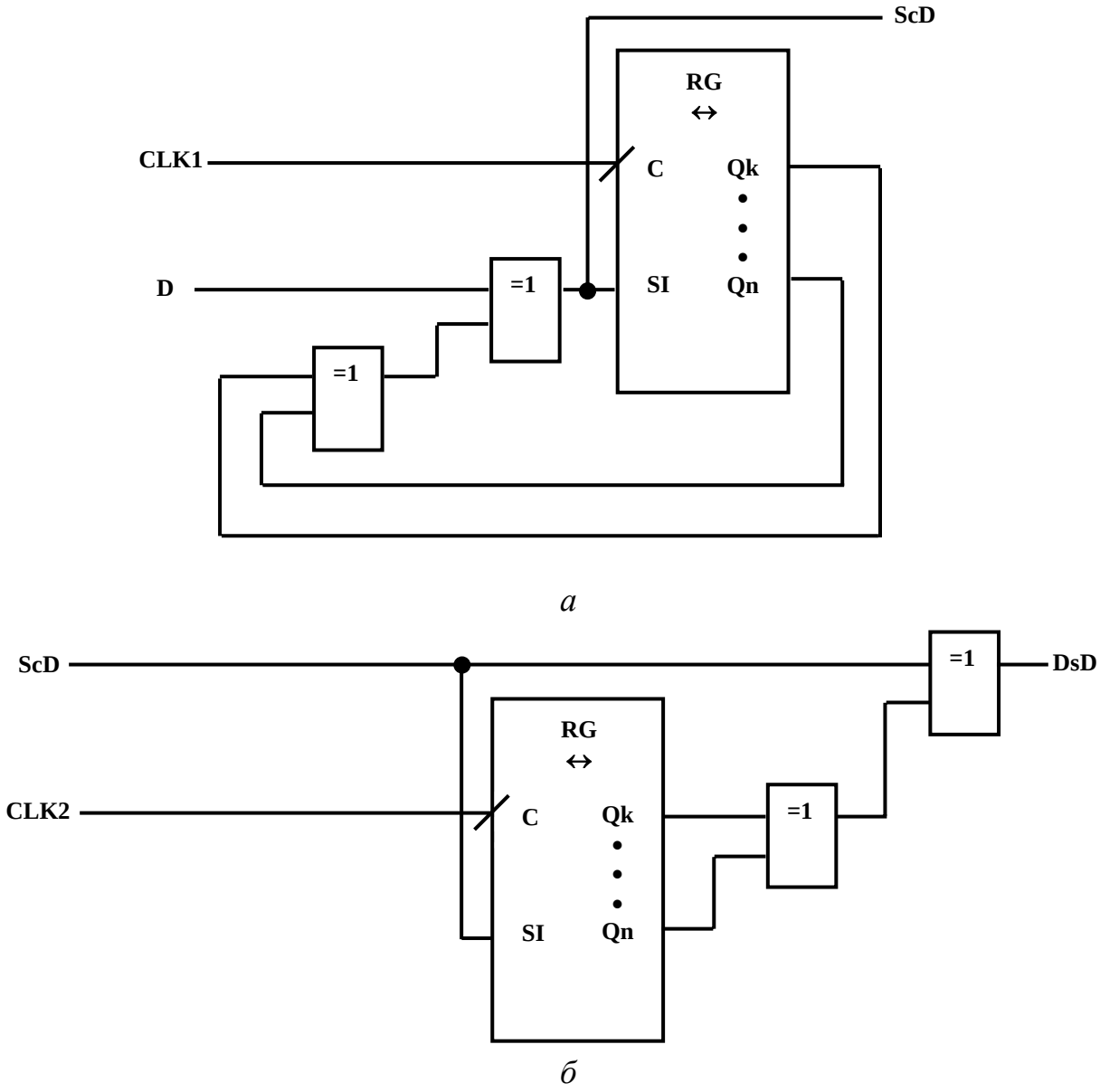
Рис. 2.32. Типовая операционная модель ГПП

Основой ГПП, как видно из рис. 2.32, является сдвиговый регистр, на вход которого поступает сумма по модулю 2 сигналов с его некоторых *k*-го и *n*-го выходов, в качестве одного из которых обычно выступает выход старшего разряда регистра. Математически такой ГПП однозначно описывается его *образующим полиномом*, имеющим вид:

$$G(x) = 1 + x^{-k} + x^{-n}. \qquad (2.23)$$

Устройство (или программа), реализующее процедуру скремблирования двоичной последовательности, называется *скремблером***,** а обратную ей процедуру восстановления исходного двоичного кода из

скремблированной последовательности – *дескремблером*. Наиболее распространенным на практике является скремблирование – дескремблирование *с самосинхронизацией* [1]. Операционные модели самосинхронизирующихся скремблера и дескремблера представлены на рис. 2.33, *а* и 2.33, *б* соответственно.



а

б

D – исходная двоичная последовательность; *ScD*, *DsD* – соответственно скремблированная и дескремблированная последовательность

Рис. 2.33. Пояснения к процессу скремблирования / дескремблирования с самосинхронизацией: операционные модели скремблера (*а*) и дескремблера (*б*)

Практическая реализация данных операционных моделей осуществляется как в аппаратной, так и (чаще) в программной формах.

Как видно из рис. 2.33, *а*, скремблированная двоичная последовательность в некотором *i*-м такте описывается следующим выражением:

$$ScD[i] = D[i] \oplus ScD[i-k] \oplus ScD[i-n]. \qquad (2.24)$$

В свою очередь, как можно увидеть из рис. 2.33, *б*, выражение, описывающее дескремблированную последовательность в *i*-м такте, имеет следующий вид:

$$DsD[i] = ScD[i] \oplus ScD[i-k] \oplus ScD[i-n]. \qquad (2.25)$$

После подстановки (2.24) в выражение (2.25) и с учетом того, что $ScD[i-k] \oplus ScD[i-k] = ScD[i-n] \oplus ScD[i-n] = 0$, получаем, что $DsD[i] = D[i]$.

Таким образом, при условии, что образующий полином ГПП скремблера совпадает с таковым ГПП дескремблера, дескремблированная последовательность полностью совпадает с исходным двоичным кодом. При этом его корректное восстановление имеет место независимо от начальных состояний сдвиговых регистров скремблера и дескремблера (при несовпадении которых корректное дескремблирование восстанавливается максимум через *n* тактов). Таким образом, рассмотренный способ скремблирования – дескремблирования обладает свойством *самосинхронизации*, т. е. не требует взаимной синхронизации моментов начальной установки указанных регистров (которая практически невозможна при территориальном удалении скремблера и дескремблера, что имеет место в ВС). Необходимо при этом отметить, что данный способ скремблирования – дескремблирования требует синхронизации моментов считывания битов скремблированной последовательности с серединами ее тактовых интервалов. Однако практическая реализация этой процедуры не вызывает серьезных трудностей (см. п. 2.4). Поэтому скремблирование – дескремблирование с самосинхронизацией достаточно широко распространено на практике.

Кроме вышерассмотренного, известен способ скремблирования – дескремблирования с синхронной начальной установкой сдвиговых регистров [1, 3]. Однако его практическая реализация в ФКС ВС весьма затруднительна. Поэтому данный способ скремблирования не нашел широкого применения в ВС.

Скремблирование, как указано ранее, позволяет практически полностью удовлетворять условиям (2.14) и (2.15). С другой стороны, оно

не позволяет обнаруживать ошибки обмена данными. Поэтому скремблирование обычно применяется в сочетании с *помехоустойчивым кодированием* на канальном или на физическом уровне, например, сверточным кодированием (подп. 2.7.4). В частности, применение скремблирования в сочетании со сверточным кодированием исходной двоичной последовательности оговаривается протоколом модуляции V.34 [7]. Скремблирование в сочетании с помехоустойчивым кодированием канального уровня используется рядом технологий ЛВС [3].

2.7.4. Сверточное кодирование

Являясь одним из методов *помехоустойчивого кодирования*, в ФКС ВС сверточное кодирование применяется для повышения устойчивости процесса обмена данными к ошибкам, обусловленным ограниченной полосой пропускания ФКС и шумами в нем. Однако сверточное кодирование, в общем случае, не позволяет удовлетворять условиям (2.14) и (2.15). Поэтому на практике подвергнутая ему двоичная последовательность с целью обеспечения этих условий обычно подвергается предварительному скремблированию или (реже) RLL-кодированию.

Подробное рассмотрение основ помехоустойчивого кодирования будет представлено далее, в гл. 3. Здесь же необходимо остановиться на них только в той мере, в которой это необходимо для понимания сущности сверточного кодирования. Основное отличие *помехоустойчивых кодов* от обычных двоичных кодов, применяемых для представления информативных данных и называемых *простыми* или *примитивными* [1, 5], состоит в следующем. *Простой код* с некоторой разрядностью N бит использует для представления информации все возможные 2^N битовых комбинаций. В *помехоустойчивом* коде с такой же разрядностью только 2^K возможных комбинаций нулей и единиц используется для представления данных (где K – некоторое целое число, меньшее N), причем соотношение между K и N определяется конкретным типом помехоустойчивого кода, а также значением N.

Используемые 2^K битовые комбинации фактически несут информацию о K битах данных и называются *разрешенными*, а неиспользуемые $2^N - 2^K$ комбинации – *запрещенными*. Их поступление на вход

приемника данных служит признаком ошибок передачи, требующим специальных мер для их исправления. В простейшем случае, оно реализуется путем запроса повторной передачи искаженного слова. Однако многие разновидности помехоустойчивых кодов, в том числе большинство сверточных кодов, позволяют, с помощью специальных алгоритмов *помехоустойчивого декодирования*, зависящих от конкретного типа кода, восстановить информативную K-битовую кодовую комбинацию из искаженного N-разрядного слова. Таким образом, за счет введения *избыточных битов* в представляемые данные обеспечивается возможность обнаружения, а в ряде практических случаев – и исправления ошибок обмена информацией. При этом максимальное количество обнаруживаемых / исправляемых ошибочных битов в N-разрядном слове зависит от конкретной разновидности помехоустойчивого кода.

Общий принцип помехоустойчивого кодирования состоит в следующем. Подвергаемая ему битовая последовательность разбивается на фрагменты разрядностью K бит. Каждый из них затем преобразуется в двоичное слово с некоторой разрядностью N, большей K. При этом соотношение между значениями K и N, а также правила преобразования определяются конкретной разновидностью помехоустойчивого кода. На приемной стороне по специальным алгоритмам, также зависящим от применяемого способа помехоустойчивого кодирования, оценивается наличие или отсутствие ошибок в принятом N-битовом слове, и из него восстанавливается представленный этим словом K-битовый фрагмент исходной двоичной последовательности или запрашивается повторная передача слова (при обнаружении ошибок, но невозможности их достоверного исправления).

Известны два основных класса помехоустойчивых кодов – *блочные* и *сверточные* [1, 5]. *Блочные коды* характеризуются тем, что биты каждого из их N-разрядных слов являются функциями *только* от K информационных битов, представляемых соответствующим словом, которое при этом называется *блоком*. *Сверточные коды* отличаются от блочных тем, что обладают *памятью*, т. е. биты их некоторого j-го N-разрядного слова являются функциями от битов как j-го K-битового фрагмента исходной двоичной последовательности, так и, в общем случае, M предыдущих K-разрядных фрагментов этой последовательности и M предыдущих N-битовых фрагментов результата кодирования. Здесь M – некоторое целое положительное число, зависящее от конкретной разновидности сверточного кода.

Значение $M+1$ называется при этом *длиной кодового ограничения* (по-англ. – *constraint length*). Сверточный код с разрядностью кодового слова N, разрядностью соответствующего ему фрагмента исходной двоичной последовательности K и длиной кодового ограничения $M+1$ в литературе часто обозначается как сверточный код (N, K, $M+1$), например, код (2, 1, 3) (см. далее).

Обычно сверточное и блочное помехоустойчивое кодирование взаимно дополняют друг друга. Образно говоря, они являются соответственно первым и вторым «рубежами обороны» в борьбе с ошибками обмена данными по ФКС.

Блочное кодирование характеризуется следующими основными особенностями [1, 5]:

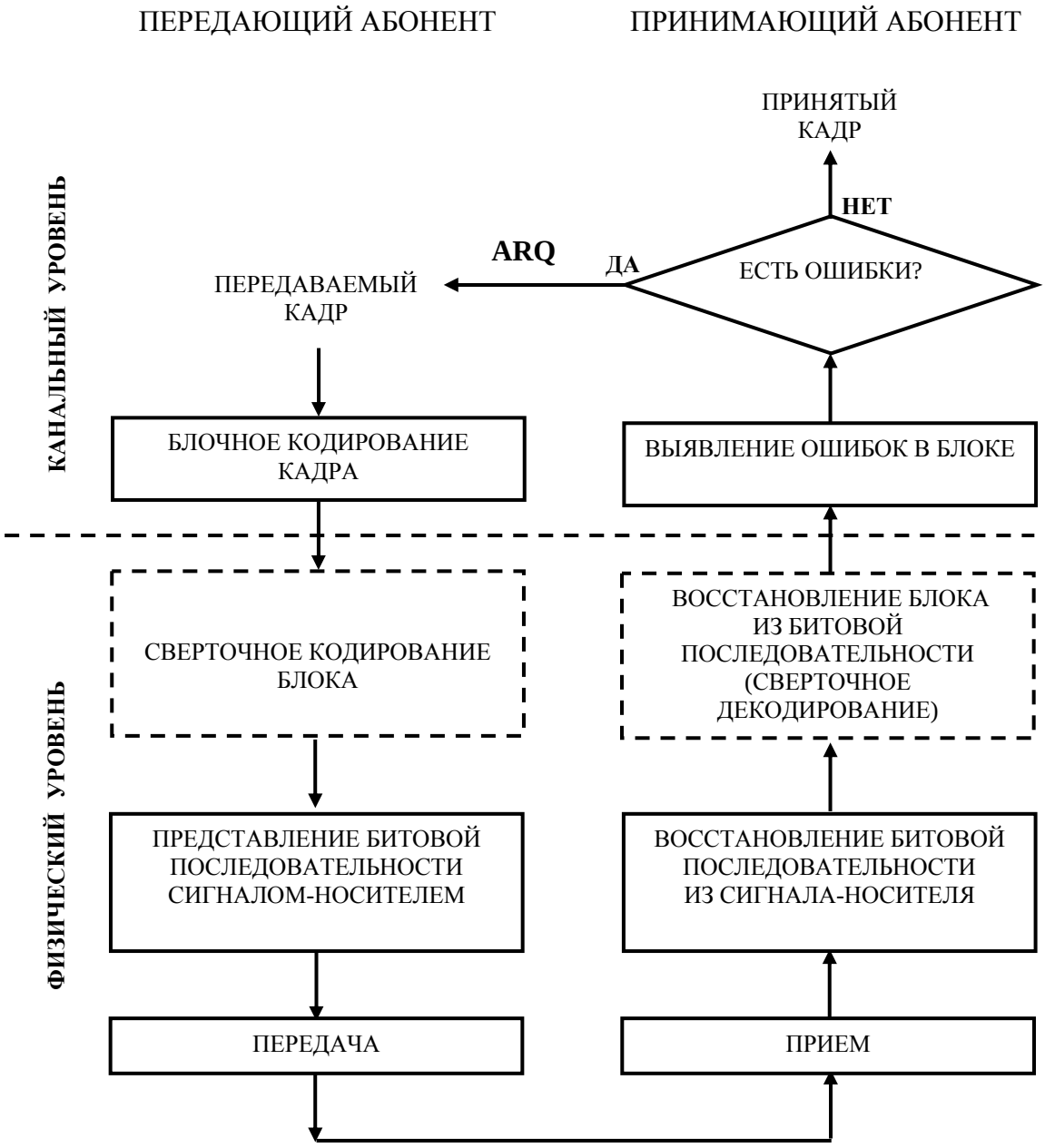
- позволяет достаточно эффективно *обнаруживать* ошибки в блоке на приемной стороне при сложности или невозможности их достоверной *коррекции*, т. е. восстановления;
- наибольшая эффективность блочного кодирования достигается при общей разрядности N блока (см. выше) от нескольких сотен бит до единиц килобит; при большей его разрядности возрастает вероятность пропуска ошибок, а при меньшей – соотношение между числом представляемых блоком информативных битов и общей разрядностью блока является неоптимально малым.

Так, например, при кодировании блочным кодом БЧХ [1, 5] для обнаружения до 4-х ошибок в блоке разрядностью 15 бит в нем должно содержаться 7 информационных и 8 избыточных (контрольных) битов, а при разрядности блока, равной 255 битам – 239 информационных и 16 контрольных битов соответственно. Таким образом, в первом случае информационные биты составляют 46,7 % от общей разрядности блока, а во втором – 93,7 %.

В свою очередь, сверточное кодирования позволяет *восстанавливать* на приемной стороне исходные данные из содержащей ошибки двоичной последовательности, при условиях, что разрядность N кодового слова не превышает нескольких бит, а ошибки имеют характер искажений одного (реже – двух или более) бита в слове.

Как следует из перечисленных свойств блочного и сверточного кодирования, рациональной (и реально применяемой) является обобщенная операционная модель помехоустойчивого кодирования/декодирования данных в ФКС ВС, представленная на рис. 2.34. Блоки, обозначенные на нем пунктирной линией, могут отсутствовать в ряде

практических случаев, так как некоторые типы ФКС ВС и способов модуляции не требуют применения сверточного кодирования (см. далее).



ARQ – автоматический запрос повторной передачи
(Automatic Repetition reQuest)

Рис. 2.34. Обобщенная операционная модель помехоустойчивого кодирования/декодирования данных в ФКС ВС

Как видно из рис. 2.34, кадр канального уровня (см. рис. 1.6 и пояснения к нему, а также гл. 3) вначале подвергается блочному помехоустойчивому кодированию. Затем сформированная в его результате двоичная последовательность (блок), в общем случае, подвергается

сверточному кодированию, преобразуется в сигнал-носитель и передается принимающему абоненту. На приемной стороне из сигнала-носителя восстанавливается представляемая им битовая последовательность, которая может не совпадать с переданной последовательностью из-за искажений сигнала-носителя в передающей среде. Затем осуществляются выявление и коррекция (в общем случае – частичная) ошибок в принятой двоичной последовательности, реализуемые в два этапа при наличии сверточного кодирования и в один этап – при его отсутствии.

При наличии сверточного кодирования первым этапом является сверточное декодирование, в процессе которого восстанавливается кадр канального уровня, закодированный блочным кодом. При восстановлении корректируются (как правило, частично) биты, искаженные шумами и помехами в ФКС и/или его ограниченной полосой пропускания. Полученный в результате декодирования блок данных также может содержать искаженные биты, что обусловлено ограниченными возможностями коррекции ошибок при сверточном декодировании. Однако наличие остаточных ошибок в блоке выявляется на втором этапе помехоустойчивого декодирования – при блочном декодировании, т. е. на основе анализа состояний битов кадра, закодированного блочным кодом. Блочное кодирование / декодирование, как указано ранее, позволяет достаточно эффективно обнаруживать ошибки в блоке, однако, как правило, не обеспечивает их достоверной коррекции на приемной стороне. Поэтому при выявлении ошибок на этапе блочного кодирования принимающий абонент посылает передающему запрос повторной передачи кадра (см. рис. 2.34).

Описанная комбинация сверточного и блочного кодирования позволяет рационально решить проблему помехоустойчивого обмена данными между абонентами ФКС ВС при относительно высоких значениях BER ФКС (см. подп. 2.1.3), порядка 10^{-3} – 10^{-4}. Это имеет место, например, в беспроводных ФКС, а также в АЛС КТСОП при использовании скоростных, но не помехоустойчивых методов модуляции (КАМ и ИКМ). В самом деле, при указанных порядках BER применение только блочного кодирования (обеспечивающего эффективное обнаружение ошибок, но обычно не позволяющего достоверно их исправлять) привело бы к недопустимому снижению скорости обмена из-за необходимости частого повторения передачи кадров, в которых обнаружены ошибки. С другой стороны, использование только сверточного кодирования привело бы к недопустимо высокому количе-

ству пропущенных ошибок в принятом кадре из-за ограниченных возможностей сверточных кодов по обнаружению и исправлению искаженных битов. Таким образом, сверточное и блочное помехоустойчивое кодирование рационально дополняют друг друга при относительно высоких уровнях BER, обусловленных шумами в ФКС или/и его ограниченной полосой пропускания.

Следует также отметить, что при относительно небольших изначальных значениях BER, порядка 10^{-5} и менее, помехоустойчивый обмен данными, как правило, обеспечивается посредством только блочного кодирования, без применения сверточного. При таких порядках BER частота повторений передачи кадров (из-за обнаружения ошибок в них) невысока, и эти повторения не приводят к существенному снижению скорости обмена. В частности, только блочное помехоустойчивое кодирование, без его дополнения сверточным, применяется в кабельных ФКС ЛВС, а также в АЛС КТСОП при использовании низкоскоростных, но помехоустойчивых протоколов модуляции, например, двухчастотной FSK, DBPSK и DQPSK (см. табл. 2.6) [1, 3, 5].

Блочное помехоустойчивое кодирование относится к операциям канального уровня, а сверточное – физического (см. рис. 2.34). Поэтому принципы и базовые алгоритмы блочного кодирования будут рассмотрены в следующей главе, посвященной сетевым технологиям канального уровня. Основы сверточного кодирования, естественно, следует рассмотреть в данном подпункте.

Обобщенное уравнение сверточного кодирования имеет следующий вид [1, 5]:

$$y_i[j] = \left\{ \sum_{k=0}^{K-1} \sum_{m=0}^{M} a_{ikm} x_k[j-m] \right\} \oplus \left\{ \sum_{n=0}^{N-1} \sum_{m=1}^{M} b_{inm} y_n[j-m] \right\}, \quad (2.26)$$

где $y_i[j]$ и $y_n[j-m]$ – соответственно i-й бит j-го и n-й бит $j-m$-го N-разрядного слова сверточного кода ($i = 0,..., N-1$);

$x_k[j-m]$ – k-й бит $j-m$-го K-разрядного слова исходной двоичной последовательности ($k = 0,..., K-1$);

a_{ikm} и b_{inm} – коэффициенты, зависящие от конкретной разновидности сверточного кода, значения которых могут быть равны 0 или 1;

$M+1$ – длина кодового ограничения.

При этом суммирование в выражении (2.26) осуществляется *по модулю 2*.

Таким образом, биты некоторого j-го N-разрядного двоичного слова, получаемого в результате сверточного кодирования, в общем случае, являются функциями от битов j-го и M предшествующих ему K-разрядных слов исходной двоичной последовательности, а также M предшествующих ему N-разрядных слов результата кодирования. Коды, обладающие такими свойствами, называют *рекурсивными* сверточными кодами. На практике до недавнего времени в основном применялись (и широко используются и в настоящее время) *нерекурсивные* коды, для которых характерна зависимость результата кодирования только от исходной двоичной последовательности, т. е. у которых все коэффициенты b_{inm} равны нулю. Другими словами, в нерекурсивном сверточном кодере отсутствуют обратные связи.

Одним из простейших примеров *нерекурсивного* сверточного кода является нерекурсивный код (2, 1, 3) [1]. Он характеризуется заменой каждого из битов исходной двоичной последовательности 2-битовым двоичным словом, формируемым в соответствии со следующими выражениями:

$$\left.\begin{aligned} y_0[j] = x_0[j] \oplus x_0[j-1] \oplus x_0[j-2]; \\ y_1[j] = x_0[j] \oplus x_0[j-2]; \end{aligned}\right\}, \qquad (2.27)$$

т. е. коэффициенты a_{000}, a_{001}, a_{002}, a_{100} и a_{102} нерекурсивного сверточного кода (2, 1, 3) равны единице, остальные коэффициенты a_{ikm} и все коэффициенты b_{inm} – нулю, а длина кодового ограничения – трем. Некоторое j-е 2-битовое слово кода (2, 1, 3) является функцией от трех 1-битовых слов (j-го и 2-х предшествующих ему) исходной двоичной последовательности.

Простейшим примером *рекурсивного* сверточного кода является рекурсивный код (2, 1, 4) [1, 5]. Для него характерно разбиение подвергаемой кодированию последовательности на 1-битовые двоичные слова, каждое из которых заменяется на 2-битовое слово, формируемое в соответствии со следующими выражениями:

$$\left.\begin{aligned} y_0[j] = x_0[j] \oplus x_0[j-1] \oplus x_0[j-3] \oplus y_0[j-2] \oplus y_0[j-3]; \\ y_1[j] = x_0[j]; \end{aligned}\right\}, \qquad (2.28)$$

т. е. коэффициенты a_{000}, a_{001}, a_{003}, a_{100}, b_{002} и b_{003} рекурсивного кода (2, 1, 4) равны единице, остальные коэффициенты a_{ikm} и b_{inm} – нулю, а длина кодового ограничения – четырем.

Сопоставляя выражения (2.27) и (2.28), можно заметить следующее. В N-битовом (где N равно 2-м) слове рекурсивного кода (2, 1, 4) в явном виде присутствует соответствующее ему K-битовое ($K=1$) слово исходной двоичной последовательности. Обладающие таким свойством сверточные коды называются *систематическими* [1, 5]. С другой стороны, в N-битовом слове нерекурсивного кода (2, 1, 3) исходное K-битовое слово в явном виде отсутствует. Такие сверточные коды известны под названием *несистематических* [1, 5]. Следует отметить, что большинство используемых на практике нерекурсивных сверточных кодов относится к категории несистематических, а рекурсивных – систематических кодов [1, 5].

Общие принципы рекурсивного и нерекурсивного сверточного кодирования сходны между собой, однако первое из них несколько сложнее в реализации и менее наглядно для объяснения. В дальнейшем, для простоты и наглядности изложения, принципы сверточного кодирования и декодирования будут рассматриваться в основном на примере несистематических нерекурсивных кодов. Вопросы рекурсивного сверточного кодирования будут затрагиваться лишь при необходимости.

Ввиду наличия *памяти* у сверточных кодов (см. выше), процессы сверточного кодирования и декодирования обычно удобнее описывать не таблицами кодирования или, соответственно, декодирования, а *диаграммами переходов*, подобными диаграммам состояний *конечных автоматов*, т. е. цифровых устройств с памятью. В принципе, сверточные кодеры могут описываться диаграммами, полностью аналогичными «классическим» диаграммам состояний цифровых автоматов [1, 5]. Однако более информативной формой представления процесса сверточного кодирования являются так называемые *решетчатые диаграммы*, отображающие последовательность изменений состояний кодера и его выходных сигналов. Отсюда происходит другое название описываемых решетчатой диаграммой алгоритмов сверточного кодирования – *решетчатое кодирование*, или *треллис-модуляция* (от англ. *trellis* – «решетка»), часто обозначаемое английской аббревиатурой *TCM* (*Trellis-Coded Modulation*).

Решетчатая диаграмма сверточного кода является графическим представлением правил кодирования. Она отображает возможные переходы между соседними по времени состояниями кодера, т. е. между его состояниями в некотором j-м и следующем за ним $j+1$-м тактах кодирования. При этом под *состоянием* сверточного кодера

в j-м такте понимается комбинация M K-битовых фрагментов исходной двоичной последовательности, предшествующих некоторому ее j-му K-разрядному фрагменту.

Правила построения решетчатой диаграммы сверточного кода иллюстрируются представленной на рис. 2.35 решетчатой диаграммой нерекурсивного кода (2, 1, 3), описываемого выражениями (2.27). Состояния кодера обозначаются на диаграмме кружками, называемыми *узлами* диаграммы, с указанием соответствующих им битовых комбинаций внутри кружка или рядом с ним. Стрелки, снабжаемые обозначениями вида $X[j]/Y[j]$ (см. рис. 2.35), указывают направления переходов между состояниями кодера в j-м и в $j+1$-м тактах при поступлении на его вход в j-м такте некоторой K-битовой комбинации $X[j]$. $Y[j]$ – выходное N-битовое слово кодера в j-м такте при подаче на его вход двоичного слова $X[j]$ и состоянии кодера, из которого исходит соответствующая стрелка. Например, согласно рис. 2.35, при состоянии *00* сверточного кодера (2, 1, 3) и поступлении на его вход единицы, на выход кодера будет выдано слово *11*, и он перейдет в состояние *10*.

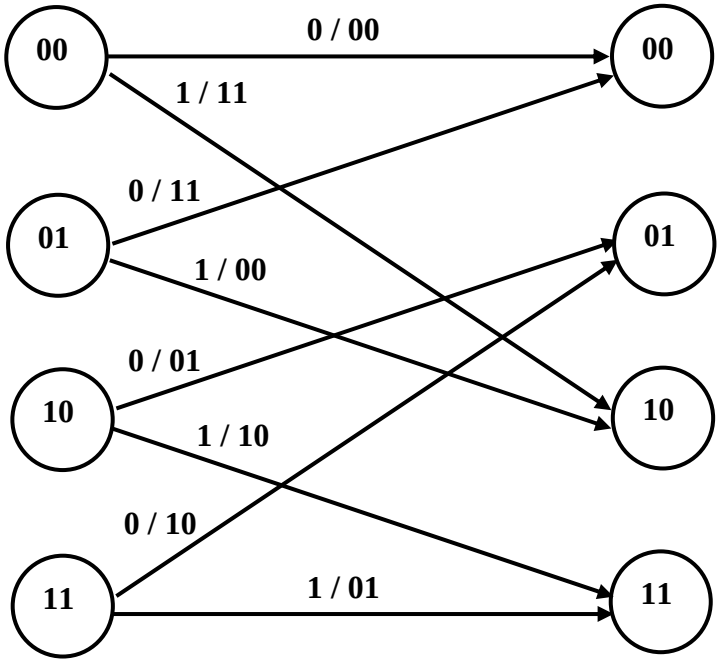


Рис. 2.35. Решетчатая диаграмма нерекурсивного сверточного кода (2, 1, 3)

Понятие решетчатой диаграммы существует и для рекурсивных кодов. Правила их построения в целом аналогичны вышеописанным.

На основе решетчатой диаграммы сверточного кода описываются процедуры сверточного кодирования и декодирования двоичных последовательностей.

Первая из названных процедур обычно представляется *трассой переходов состояний и выходов* кодера при подаче на его вход кодируемой двоичной последовательности. В качестве примера на рис. 2.36 представлена подобная трасса кодера нерекурсивного сверточного кода (2, 1, 3), описываемого приведенной на рис. 2.35 решетчатой диаграммой, при подаче на вход кодера двоичной последовательности *1001*.

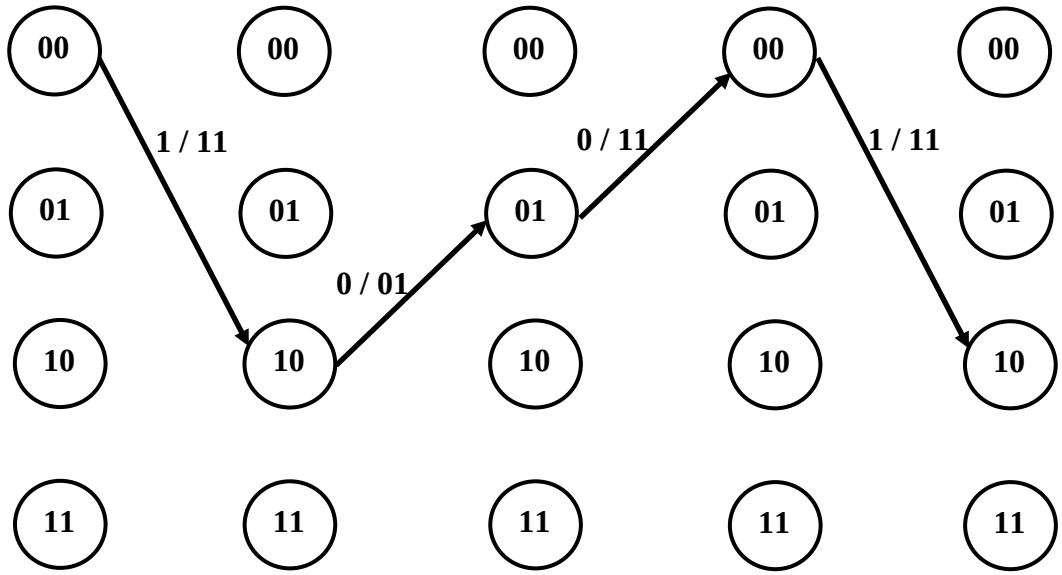


Рис. 2.36. Трасса переходов состояний и выходов кодера нерекурсивного сверточного кода (2, 1, 3) при подаче на его вход двоичной последовательности *1001* (выходная битовая комбинация кодера – *11011111)*

Необходимо отметить, что при построении трасс переходов и выходов сверточных кодеров их *начальное состояние* полагается нулевым [1] (см. рис. 2.36). В остальном рис. 2.36 не нуждается в комментариях. Из него нетрудно заметить, что при поступлении битовой комбинации *1001* на вход кодера нерекурсивного сверточного кода (2, 1, 3) в результате кодирования будет получена двоичная последовательность *11011111*.

Сверточное декодирование, т. е. восстановление исходной двоичной последовательности из сверточного кода с коррекцией ошибок передачи (в общем случае – частичной) может быть реализовано несколькими методами, большинство из которых основывается на при-

менении *трасс переходов состояний и выходов* декодера [1, 5]. На рис. 2.37 в качестве простейшего примера приведена подобная трасса декодера нерекурсивного сверточного кода (2, 1, 3) при поступлении на его вход *разрешенной* двоичной последовательности *11011111* (см. рис. 2.36 и пояснения к нему). Данная трасса построена на основе решетчатой диаграммы указанного кода (см. рис. 2.35). Переходы между состояниями декодера определяются очередным N-битовым словом сверточного кода (рис. 2.37), а не как при кодировании очередной K-битовой комбинацией исходной двоичной последовательности (см. рис. 2.36), которая при декодировании изначально неизвестна, и определяется на основании текущего состояния декодера, а также соответствующего ему входного N-битового слова. Начальное состояние декодера, как и кодера, всегда полагается нулевым (рис. 2.37).

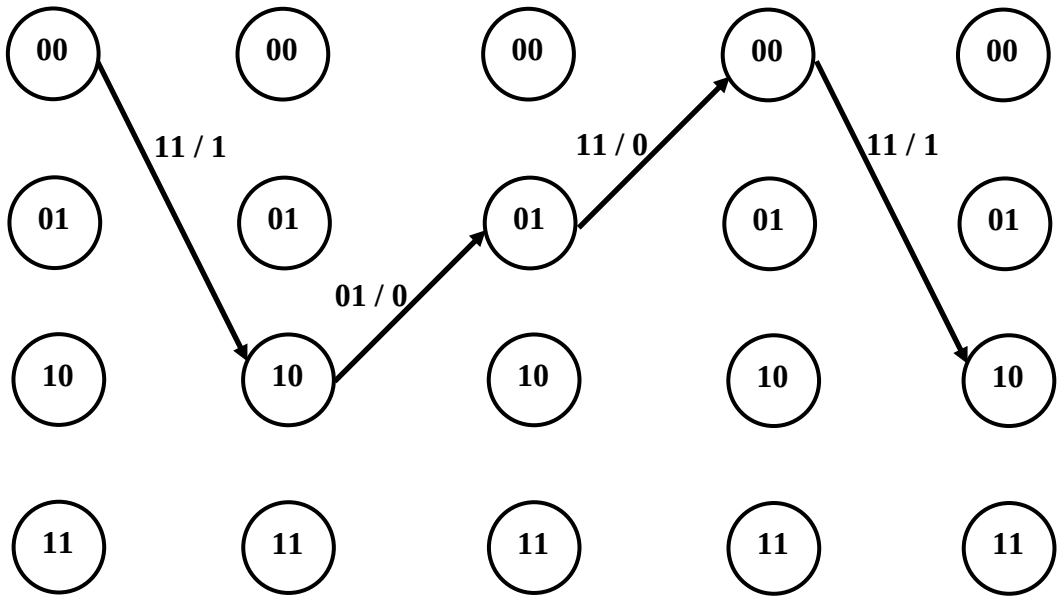


Рис. 2.37. Трасса переходов состояний и выходов декодера нерекурсивного сверточного кода (2, 1, 3) при подаче на его вход разрешенной двоичной последовательности *11011111* (выходная битовая комбинация декодера – *1001*)

В общем случае, поступающая на вход декодера кодовая комбинация может содержать ошибки, а задача сверточного декодирования заключается в максимально достоверном восстановлении исходной битовой комбинации из входной двоичной последовательности декодера. При этом наличие или отсутствие в ней искаженных битов, а также их позиции изначально «неизвестны» на приемной стороне.

Известен ряд алгоритмов такого восстановления [1, 5]. Большинство из них реализует принцип *максимального правдоподобия*. Он заключается в том, что при декодировании на основе полученной на приемной стороне двоичной последовательности восстанавливается наиболее вероятная (в соответствии с критериями, используемыми конкретным алгоритмом) трасса переходов состояний и выходов кодера.

Наиболее простым по сущности (но не с точки зрения аппаратно-программных затрат на реализацию) из алгоритмов этой группы является описываемый следующей последовательностью действий [1].

1. На основании решетчатой диаграммы соответствующего сверточного кода строится совокупность *всех* возможных трасс переходов состояний и выходов декодера при разрядности входной двоичной последовательности, равной разрядности декодируемой битовой комбинации. Они строятся так же, как и трасса, представленная на рис. 2.37, на основе решетчатой диаграммы соответствующего кода.

2. Для каждой из построенных трасс вычисляется *метрика ветвлений* (в дальнейшем – *метрика*). Она определяется как сумма различий между значениями битов реально принятой декодером двоичной последовательности и одноименных битов его входной последовательности, при которой смена состояний декодера описывалась бы соответствующей трассой.

3. Трасса с минимальной метрикой, т. е. соответствующая входной последовательности N-битовых слов декодера с минимальным отклонением от реально принятой, полагается наиболее правдоподобным повторением трассы переходов состояний и выходов кодера. По ней восстанавливается соответствующая ей последовательность K-битовых слов, служащая результатом декодирования. В общем случае, из-за ограниченных корректирующих возможностей сверточных кодов, она может содержать ошибки, т. е. не совпадать с входной двоичной последовательностью сверточного кодера передающего абонента. Эти ошибки обнаруживаются и корректируются на канальном уровне (см. рис. 2.34).

Следует также отметить, что в ряде частных случаев критерию минимальной метрики может удовлетворять не одна, а несколько трасс. При этом для выбора максимально правдоподобной из них необходимо применение специальных алгоритмов [1, 5], рассмотрение которых выходит за рамки настоящего учебного пособия.

Вышеописанный алгоритм иллюстрируется рис. 2.38. На нем представлена совокупность возможных трасс переходов состояний декодера нерекурсивного сверточного кода (2, 1, 3), описываемого решетчатой диаграммой (см. рис. 2.35), при подаче на вход декодера 8-битовой двоичной последовательности.

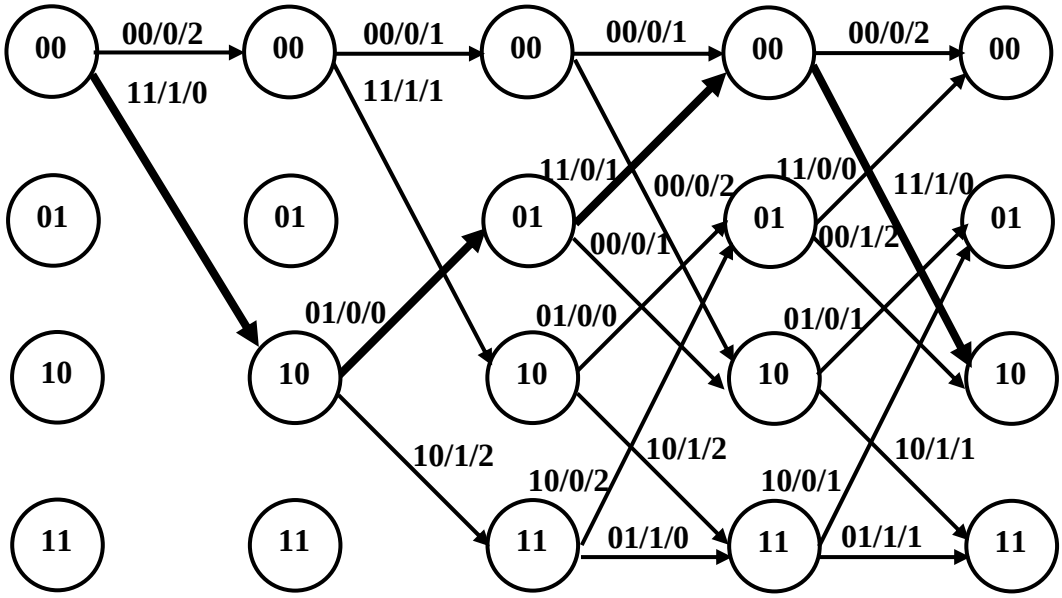


Рис. 2.38. Пример реализации сверточного декодирования по принципу максимального правдоподобия (код – нерекурсивный (2, 1, 3), описываемый рис. 2.35; входная последовательность – *11010111*; результат декодирования – *1001*)

При этом предполагается, что реально на его вход поступила *запрещенная* битовая комбинация *11010111*, представляющая собой двоичную последовательность *11011111* с одним искаженным битом. При этом рядом с каждой из стрелок, составляющих трассы, указана, кроме соответствующих ей (т. е. обозначаемому этой стрелкой переходу) N-битовой входной и K-битовой выходной комбинации декодера (см. рис. 2.37), также *метрика* данной стрелки. Она, аналогично метрике трассы (см. выше), определяется как сумма различий между значениями битов N-разрядной входной двоичной последовательности декодера, которая вызвала бы описываемую данной стрелкой смену его состояния, и значениями одноименных битов реально принятого декодером N-разрядного слова, соответствующего указанной смене состояния по его позиции в декодируемой последовательности. Метрика трассы вычисляется как сумма метрик составляющих ее стрелок.

Из рис. 2.38 видно, что наименьшим значением метрики и, следовательно, максимальным правдоподобием обладает трасса, обозначенная стрелками большей интенсивности. Поэтому в соответствии с рассматриваемым алгоритмом принимается решение о том, что при поступлении запрещенной двоичной последовательности *11010111* на вход декодера нерекурсивного сверточного кода (2, 1, 3), наиболее вероятным является отправление передатчиком приемнику битовой комбинации *11011111*, что и имело место на самом деле. Она, в свою очередь, представляет собой результат сверточного кодирования двоичного слова *1001* (см. рис. 2.36 и 2.38), которое при этом и служит результатом декодирования.

В рассмотренном примере результат декодирования не содержит ошибок. В общем случае, как указано выше, он может содержать остаточные ошибки, устраняемые на канальном уровне.

Рассмотренный выше алгоритм является простым по сущности, но «неэкономным» с точки зрения затрат памяти и времени на его реализацию из-за необходимости анализа *всех* возможных трасс переходов состояний и выходов декодера. Поэтому на практике обычно применяются другие алгоритмы сверточного кодирования, также основанные на принципе максимально правдоподобного восстановления трассы переходов состояний декодера, но более «экономные». Наиболее распространенными из них являются различные варианты *алгоритма Витерби* [1, 5]. Их основным отличием от вышеописанного алгоритма является исключение заведомо наименее правдоподобных трасс из процесса дальнейшего анализа на каждом шаге декодирования. Один из простейших вариантов алгоритма Витерби основывается на исключении указанных трасс из рассмотрения по следующему правилу. Если на некотором шаге декодирования в какой-либо узел входит две или несколько различных трасс, то из дальнейшего анализа исключаются все из них, кроме обладающей наименьшей среди них метрикой. Если существует несколько трасс, удовлетворяющих этому критерию, то, по простейшему варианту алгоритма, оставляются все они. При этом трассы, оставшиеся на последнем шаге декодирования после исключения всех наименее правдоподобных, называются *«выжившими»*. Трасса, обладающая минимальной метрикой среди «выживших», полагается наиболее правдоподобной. Как и при использовании ранее рассмотренного алгоритма (см. рис. 2.38), критерию минимальной метрики могут удовлетворять несколько из «выживших» трасс. Наиболее правдоподобная из них выбирается по специальным алгоритмам.

Вышеописанный простейший вариант алгоритма Витерби иллюстрируется рис. 2.39.

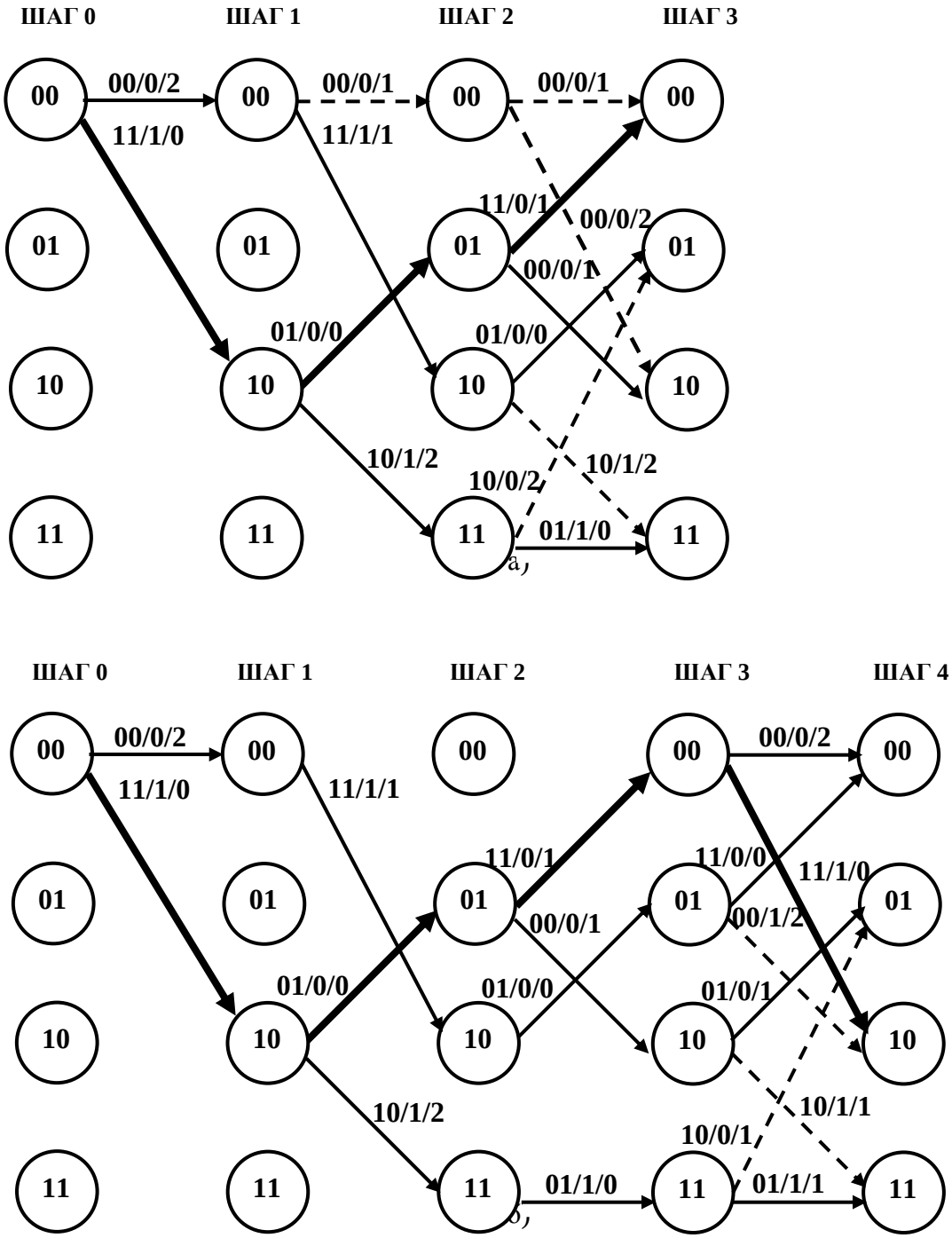


Рис. 2.39. Пояснения к процессу сверточного декодирования по алгоритму Витерби (код – (2, 1, 3), описываемый рис. 2.35; входная последовательность – *11010111*; результат декодирования – *1001*; пунктиром обозначены исключаемые из рассмотрения трассы)

На нем представлены трассы переходов состояний и выходов декодера нерекурсивного сверточного кода (2, 1, 3) на 3-м и 4-м шагах

декодирования, с указанием исключаемых трасс пунктирными стрелками. На 0-м, 1-м и 2-м шагах, как нетрудно заметить из рис. 2.39, трассы, подлежащие исключению в соответствии с вышеприведенным правилом, еще не могут быть выявлены. Как следует из рис. 2.39, минимальной метрикой среди «выживших» характеризуется трасса, отмеченная стрелками большей интенсивности. Ей соответствует результат декодирования *1001*, совпадающий с полученным по ранее рассмотренному алгоритму (см. рис. 2.38).

Нетрудно заметить, что алгоритм Витерби характеризуется меньшими затратами памяти и временем анализа, чем ранее описанный алгоритм, за счет исключения заведомо неправдоподобных трасс. С другой стороны, он более сложен в реализации (что, однако, не является серьезным недостатком при существующем уровне развития аппаратно-программных средств кодирования/декодирования). Поэтому алгоритм Витерби достаточно широко применяется на практике [1, 5]. Как и иллюстрируемый рис. 2.38 алгоритм, он не гарантирует отсутствия остаточных ошибок в декодированной двоичной последовательности, которые обнаруживаются и устраняются на канальном уровне (см. рис. 2.34).

Кроме вышерассмотренных, естественно, существуют и другие алгоритмы сверточного декодирования (в том числе более «развитые» варианты алгоритма Витерби) [1, 5].

Декодирование рекурсивных сверточных кодов в целом реализуется по тем же общим принципам, что и нерекурсивных, в том числе на основе принципа максимального правдоподобия. Однако конкретные алгоритмы декодирования рекурсивных кодов отличаются от применяемых для нерекурсивных кодов [1, 5]. В частности, на практике достаточно широко распространены алгоритмы группы *MAP* (от англ. аббревиатуры «*Maximum A Posteriori*», в переводе – *максимальная апостериорная вероятность*) [1, 5].

В целом, сверточное кодирование, при прочих равных условиях, позволяет осуществлять корректный обмен данными при отношении «сигнал-шум» в канале связи, на 3 – 6 дБ меньшем, чем в отсутствие сверточного кодирования. Кроме того, его применение не требует строгого удовлетворения условию (2.13) ввиду возможности коррекции ошибок, обусловленных его несоблюдением. Это позволяет обеспечить большую скорость обмена данными, чем ограничиваемая

условием (2.13). Следует также отметить, что примерно до середины 90-х гг. прошлого века на практике применялись, в основном нерекурсивные несистематические сверточные коды. Рекурсивные коды используются в основном в более современных протоколах логического кодирования, в частности, при турбо-кодировании (см. далее).

Основными *недостатками* «классических» сверточных кодов, описываемых выражением (2.26), являются:

- увеличение разрядности исходной двоичной последовательности при кодировании в N / K раз, например, при (2, 1, 3)-кодировании – в 2 раза, что приводит к снижению эффективной скорости обмена данными в такое же количество раз, так как N-битовое двоичное слово сверточного кода реально несет информацию только о K битах исходной последовательности (в то время как для используемых на практике блочных кодов характерно снижение скорости обмена на 10 – 20 % [1, 5]);

- ограниченные возможности или невозможность достоверной (т. е. без остаточных ошибок) коррекции *групповых искажений битов*, т. е. следующих подряд двух или нескольких искаженных битов (что весьма часто имеет место в реальных ФКС).

Однако современные методы и алгоритмы сверточного кодирования позволяют частично устранить перечисленные недостатки. Основным методом устранения *первого* из них является так называемое *выкалывание* (*code puncturing*) [1]. Принцип выкалывания заключается в избирательном удалении некоторых избыточных битов из двоичной последовательности, являющейся результатом сверточного кодирования. Такой подход в ряде случаев позволяет достигнуть рационального компромисса между возможностью исправления ошибок и степенью избыточности сверточного кода (т. е. степенью увеличения разрядности результата сверточного кодирования по сравнению с исходной двоичной последовательностью).

Общий принцип реализации выкалывания состоит в следующем [1]. Позиции удаляемых битов выходной последовательности кодера задаются так называемой *матрицей выкалывания*, определяемой типом сверточного кода и конкретным протоколом кодирования. Естественно, данная матрица должна быть «известна» как на передающей, так и на приемной стороне. Число строк матрицы выкалывания равно разрядности N выходного слова кода, а число столбцов – пери-

оду выкалывания, т. е. периоду повторения позиций удаляемых битов. Ниже представлен один из примеров матрицы выкалывания, используемой в сверточном кодировании при разрядностях входного и выходного слова, равных одному и двум битам соответственно.

$$\boldsymbol{P}=\begin{pmatrix}1 & 0 & 1\\ 1 & 1 & 0\end{pmatrix}. \qquad (2.29)$$

Данная матрица соответствует следующему алгоритму выкалывания: в последовательности выходных 2-битовых слов в каждом $3j$-м слове не удаляется ни один бит, в каждом $3j+1$-м – младший бит, а в каждом $3j+2$-м – старший (где j – целое неотрицательное число). Таким образом, каждым 3-м битам входной двоичной последовательности соответствует 4 бита выходной. Поэтому говорят, что относительная скорость передачи данных после выкалывания, описываемого матрицей (2.29), равна 3/4 [1].

Следует отметить, что применяемые на практике матрицы выкалывания, в том числе вышеприведенная, составляются таким образом, чтобы минимизировать обусловленные выкалыванием ошибки декодирования (см. далее).

На приемной стороне перед декодированием в позиции удаленных при выкалывании битов заносятся некоторые значения (нули или единицы), определяемые конкретным алгоритмом декодирования. Этим битам присваивается *нулевой уровень доверия*, что учитывается при декодировании [1]. Например, они могут игнорироваться в процессе вычисления метрик ветвлений при декодировании по алгоритмам Витерби. Естественно, это снижает достоверность декодирования. Однако при рациональном выборе матрицы выкалывания и алгоритма декодирования может быть обеспечено приемлемое сочетание скорости обмена данными и вероятности ошибок декодирования [1].

Возможность коррекции *групповых искажений битов* при применении сверточных кодов улучшается при использовании процедуры *чередования*, называемого также *перемежением* (по-англ. – *interleaving*) [1]. В простейшем случае, оно состоит в следующем. На передающей стороне, после сверточного кодирования, полученная в его результате двоичная последовательность преобразуется таким образом, что ее биты переставляются местами по определенным правилам, зависящим от конкретного алгоритма чередования. В частности, на

практике достаточно широко применяется *псевдослучайная перестановка*, реализуемая в соответствии с выражением:

$$Inr[i] = Cd[Psr(i)], \qquad (2.30)$$

где $Inr[i]$ – i-й бит подвергнутой чередованию двоичной последовательности, причем $i = 0,..., L-1$, где L – ее разрядность (в битах);

$Psr(i)$ – i-й элемент последовательности целых псевдослучайных чисел, находящихся в пределах от 0 до $L-1$;

$Cd[Psr(i)]$ – бит с номером $Psr(i)$ исходной (подвергаемой чередованию) двоичной последовательности разрядностью L.

При этом псевдослучайная последовательность чисел $Psr(i)$ генерируется посредством ГПП, реализуемых в аппаратной или программной форме (см. подп. 2.7.3). Естественно, известны и другие алгоритмы чередования, например: на основе циклического сдвига исходной последовательности; четно-нечетной перестановки и ряд других [1].

После выполнения операции чередования смежные биты результата сверточного кодирования оказываются разнесенными между собой на несколько позиций, минимальное количество которых также определяется алгоритмом чередования. После процедуры чередования двоичная последовательность преобразуется в сигнал-носитель, отправляемый в передающую среду. На приемной стороне, после извлечения из сигнала-носителя представляемой им битовой последовательности, она подвергается процедуре *дечередования*, т. е. восстановления порядка следования битов, который имел место до выполнения операции чередования на передающей стороне. Например, при чередовании методом псевдослучайной перестановки дечередование осуществляется посредством преобразования, обратного описываемому выражением (2.30), с применением ГПП, идентичного используемому при чередовании. Полученная в результате дечередования двоичная последовательность затем подвергается сверточному декодированию. При этом, если в процессе передачи будут искажены несколько соседних битов представленной сигналом-носителем двоичной последовательности (вероятность чего велика), после дечередования они окажутся разнесенными между собой. Это повышает достоверность коррекции таких искажений при сверточном декодировании [1].

Вышеописанный способ чередования поясняется нижеприведенным рис. 2.40 (см. подрисуночную подпись).

Метод чередования используется в сравнительно новом классе помехоустойчивых кодов – так называемых *турбо-кодах* [1, 5]. Принцип турбо-кодирования был предложен в 1993 году. В настоящее время оно находит все более широкое применение в различных протоколах обмена данными по ФКС, постепенно заменяя «классические» помехоустойчивые коды в ряде приложений (например, в беспроводных ФКС). Основное преимущество турбо-кодов состоит в том, что, благодаря повышенной (по сравнению с ранее применявшимися помехоустойчивыми кодами) способности обнаружения и коррекции ошибок, обусловленных шумами в ФКС и его ограниченной полосой пропускания, они позволяют вплотную приблизиться к максимально возможной пропускной способности ФКС, задаваемой выражением (2.8).

Рис. 2.40. Пояснения простейшего способа чередования: операционная модель его реализации (*а*); примеры последовательностей битов, полученных в результате линейного декодирования / демодуляции (*б*) и последующего дечередования (*в*); косым крестом отмечены искаженные биты

Вообще говоря, как и для помехоустойчивых кодов в целом, существуют понятия блочных и сверточных турбо-кодов. В настоящем

подпункте будут вкратце рассмотрены принципы сверточного турбо-кодирования, относящегося к процедурам физического уровня логической модели ВС.

Принцип сверточного турбо-кодирования поясняет рис. 2.41, на котором представлена операционная модель реализующего его кодера.

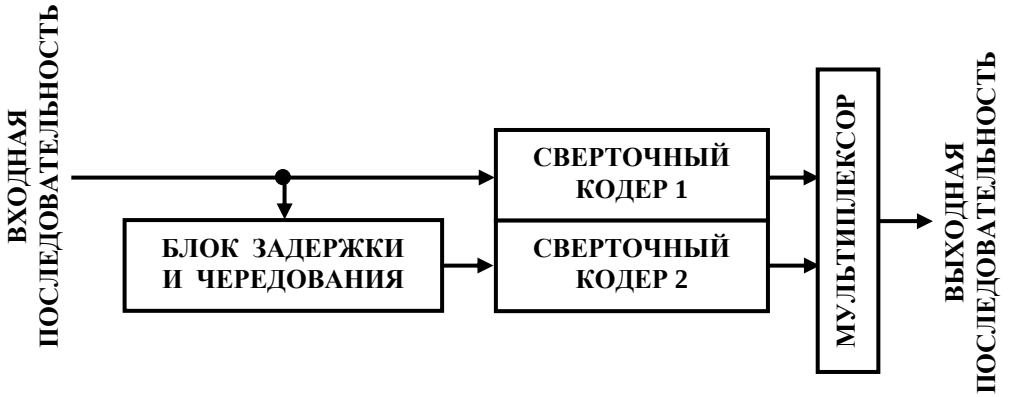


Рис. 2.41. Операционная модель сверточного турбо-кодера

Кодирование основывается на *параллельной конкатенации*, т. е. одновременном преобразовании входной двоичной последовательности в два различных битовых потока, с последующим их объединением в выходную последовательность путем мультиплексирования. Указанное преобразование осуществляется посредством 2-х параллельно работающих кодеров, реализуемых в аппаратной или программной форме. Обычно на практике они осуществляют рекурсивное систематическое кодирование и являются идентичными, т. е. описываются полностью аналогичными выражениями вида (2.26). Однако не идентичны входные данные этих кодеров: на первый из них подается непосредственно входная двоичная последовательность турбо-кодера, а на второй – та же последовательность, подвергшаяся процедурам задержки (на число тактов, определяемое конкретным алгоритмом кодирования) и чередования. Благодаря этому выходная последовательность турбо-кодера представляет собой композицию двух битовых потоков, являющихся *различными* представлениями *одной и той же* двоичной последовательности. В процессе передачи результата турбо-кодирования по ФКС в каждом из указанных потоков бит, естественно, могут возникнуть ошибки. Однако вероятность возникновения *идентичных* ошибок в обоих потоках пренебрежимо мала. Это позволяет «сопоставлять» между собой результаты декодирования каждого из них на приемной стороне и за счет этого обнаруживать

и корректировать искажения битов, которые невозможно было бы обнаружить при наличии только одного из названных битовых потоков.

Можно провести аналогию между восстановлением исходной двоичной последовательности из результата турбо-кодирования и восстановлением текста на некотором языке из выполненных независимо друг от друга переводов этого текста на два других языка, каждый из которых содержит свои неточности. Восстановление же исходной последовательности из «классического» сверточного кода аналогично восстановлению указанного текста только из одного содержащего неточности перевода. Нетрудно заметить, что в первом случае достоверность восстановления будет существенно выше.

Основным недостатком турбо-кодов является относительно высокая сложность реализации кодирования и декодирования. Однако при современном уровне развития аппаратно-программных средств обработки цифровых данных этот недостаток не является существенным. Поэтому, как указано ранее, турбо-коды находят все более широкое применение в различных сетевых протоколах физического и канального уровней. В частности, применение как сверточных, так и блочных турбо-кодов предусматривается рядом стандартов технологии беспроводного абонентского доступа WiMAX [1] (см. подп. 2.2.3).

2.7.5. Техническая реализация логического кодирования/декодирования

Как и другие функции физического уровня модели OSI, логическое кодирование осуществляется посредством сетевых адаптеров и/или модемов абонентских компьютеров, а также функционально аналогичных блоков коммутационных устройств ВС (коммутаторов, маршрутизаторов и т. п.), преимущественно в программной форме. Принципы реализации данного сетевого оборудования будут рассмотрены в гл. 3.

Выводы по п. 2.7

Из вышеизложенного материала можно сделать следующие выводы об основных свойствах и областях применения методов логического кодирования, рассмотренных в подп. 2.7.2 – 2.7.4.

Избыточное RLL-кодирование применяется преимущественно для удовлетворения условия (2.14) (в ряде практических случаев –

и (2.15)) при линейном кодировании, причем каждый конкретный тип RLL-кода, как правило, обеспечивает выполнение условий (2.14) и (2.15) только для некоторых определенных типов линейного кода и предназначен для совместного использования с ними. Например, RLL-код 4B/5B (см. табл. 2.7) используется в основном совместно с линейными кодами NRZI или MLT-3. При применении модулированных сигналов-носителей избыточное RLL-кодирование применяется сравнительно редко. Благодаря избыточности, данный тип RLL-кодирования также позволяет обнаруживать единичные искажения битов при обмене данными по ФКС (однако его возможности обнаружения ошибок достаточно ограничены по сравнению со сверточными и блочными помехоустойчивыми кодами). Необходимо также отметить, что RLL-кодирование практически не влияет на спектральный состав сигнала-носителя. Однако за счет избыточности оно несколько снижает эффективную скорость передачи данных (например, код 4B/5B – в 5/4 раза, т. е. в 1,25 раз). Таким образом, основной областью практического применения избыточного RLL-кодирования является логическое кодирование данных физического уровня в скоростных ФКС, характеризуемых использованием линейно-кодированных сигналов-носителей, и относительно невысоким уровнем BER (см. подп. 2.2.3), т. е. в скоростных кабельных ФКС ЛВС. В частности, как указано ранее, код 4B/5B применяется в технологиях ЛВС FDDI и Fast Ethernet [3].

Скремблирование обеспечивает удовлетворение условий (2.14) и (2.15) практически для любой двоичной последовательности [1, 3], представляемой как линейно-кодированным, так и модулированным сигналом. Важным свойством скремблирования является также отсутствие избыточности, влияния на граничные частоты спектра сигнала-носителя и на скорость обмена данными. С другой стороны, скремблирование не обеспечивает обнаружения искаженных битов. Исходя из вышесказанного, основными областями применения скремблирования является логическое кодирование исходной двоичной последовательности при ее представлении модулированным сигналом, а также ряд скоростных технологий ЛВС [3].

Сверточное кодирование применяется для коррекции на физическом уровне ошибок обмена данными по ФКС с высоким уровнем BER (см. рис. 2.34 и пояснения к нему), преимущественно в беспроводных ФКС ВС. Как правило, сверточное кодирование не обеспечивает выполнение условий (2.14) и (2.15). Поэтому обычно оно

используется в сочетании с преобразованием исходной двоичной последовательности, обеспечивающим удовлетворение этих условий, например, скремблированием [3].

Естественно, кроме вышеописанных, известны и другие методы логического кодирования данных на физическом уровне [1, 5]. В частности, существуют специальные методы кодирования, используемые при некоторых методах мультиплексирования (например, кодовом мультиплексировании), а также расширения спектра. Наиболее распространенные из них будут рассмотрены далее, в пп. 2.8 и 2.9 соответственно.

2.8. Мультиплексирование ФКС ВС

2.8.1. Общие положения

Как было указано в п. 2.4, мультиплексирование уплотнения заключается в использовании одного и того же участка передающей среды множеством ФКС, необходимость которого обусловлена:

- для кабельных ФКС – технической и организационно-экономической сложностью прокладки отдельного участка кабеля для каждого из ФКС во многих практических случаях;

- для беспроводных ФКС – разделяемым характером беспроводной передающей среды по самой ее природе (см. подп. 2.3.4).

В свою очередь, для использования одного и того же участка передающей среды множеством ФКС возникает необходимость в специальных мерах для устранения взаимного влияния сигналов-носителей данных этих ФКС при их передаче по общему участку среды. Указанные меры реализуются посредством специальных приемов, также обычно называемых мультиплексированием.

В настоящее время известны следующие основные методы мультиплексирования ФКС ВС [1, 3, 5, 7]:

- *частотное мультиплексирование* (по-англ. – *Frequency-Division Multiplexing*, сокращенно *FDM*), частным случаем которого является *волновое мультиплексирование* (в англоязычной литературе – *Wavelength-Division Multiplexing*, *WDM*), применяемое в ФКС на основе ВОК;

- *временное мультиплексирование* (по-англ. – *Time-Division Multiplexing, TDM*);

- *кодовое мультиплексирование* (по-англ. – *Code-Division Multiple Access, CDMA*).

На практике применяются также сочетания перечисленных методов [1, 3, 5, 7].

Основы реализации основных методов мультиплексирования ФКС ВС рассмотрены в подп. 2.8.2 – 2.8.5.

2.8.2. Частотное мультиплексирование (FDM)

Данный метод состоит в выделении под сигнал-носитель данных каждого из мультиплексируемых ФКС определенного частотного диапазона, не перекрывающегося с частотными диапазонами сигналов-носителей других ФКС, использующих тот же участок передающей среды. Принцип частотного мультиплексирования иллюстрируется рис. 2.42.

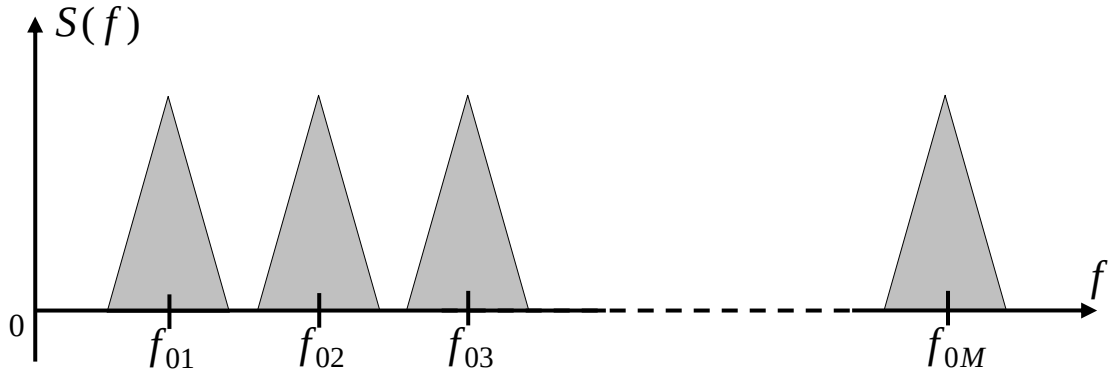


Рис. 2.42. Пример распределения СПМ сигналов-носителей данных в ФКС при частотном мультиплексировании:
f_{01}, f_{02}…, f_{0M} – частоты несущих; M – число ФКС, разделяющих один и тот же участок передающей среды; $S(f)$ – спектральная плотность мощности (СПМ) сигналов-носителей данных

Частотное мультиплексирование предполагает использование модулированных синусоидальных сигналов-носителей данных, спектры которых сосредоточены в окрестностях частоты несущей (см. выражение (2.4)). Каждому из ФКС выделяется несущая определенной ча-

стоты, выбираемой таким образом, чтобы для любой пары соседних по частоте несущих выполнялось следующее условие:

$$f_{0n} - f_{0m} > (f_{HSm} - f_{0m}) + (f_{0n} - f_{LSn}), \qquad (2.31)$$

где $n = m + 1$;

f_{0m} и f_{0n} – частоты несущих соответственно m-го и n-го ФКС;

f_{HSm} – верхняя граничная частота спектра сигнала-носителя m-го ФКС;

f_{LSn} – нижняя граничная спектра сигнала-носителя n-го ФКС.

Из рис. 2.42 нетрудно заметить, что выполнение условия (2.31) обеспечивает отсутствие взаимного перекрытия спектров сигналов-носителей мультиплексируемых ФКС и, следовательно, взаимного влияния этих сигналов.

При передаче цифровых данных по ФКС частотное мультиплексирование, как правило, предполагает *два этапа модуляции*: формирование модулированного (манипулированного) сигнала-носителя цифровых данных и последующий перенос спектра этого сигнала в диапазон частот, выделенный для конкретного ФКС.

Приемник ФКС посредством полосно-пропускающего фильтра, настроенного на диапазон частот соответствующего ФКС, выделяет сигнал-носитель, после чего осуществляются *два последовательных этапа демодуляции*: обратный перенос спектра сигнала-носителя и извлечение из него цифровых данных.

Перенос спектра на передающей стороне обычно реализуется методом амплитудной или частотной модуляции несущей, выделенной под соответствующий ФКС, сигналом-носителем цифровых данных [1, 5]. Обратный перенос спектра на приемной стороне осуществляется, соответственно, посредством амплитудного или частотного детектирования.

Формирование сигнала-носителя цифровых данных при FDM, как правило, реализуется одним из следующих двух методов [1, 5]:

- модуляцией с одной несущей (обычно называемой *поднесущей*);
- модуляцией с множеством *взаимно ортогональных* поднесущих (смысл понятия взаимной ортогональности будет пояснен далее).

Частотное мультиплексирование, основанное на первом из перечисленных методов, назовем «классическим» FDM. При нем сигнал-носитель цифровых данных формируется как результат модуляции синусоидального сигнала (поднесущей) двоичной последовательностью, подлежащей передаче. Модуляция при этом обычно осуществ-

ляется методами MSK, PSK (DPSK) или QAM (см. подп. 2.6.2 – 2.6.4), причем частота поднесущей, как правило, одинакова для всех мультиплексируемых ФКС (собственно частотное мультиплексирование осуществляется на этапе переноса спектра) [1, 5]. Данный метод формирования сигнала-носителя цифровых данных достаточно прост и не нуждается в подробных комментариях. Он применяется в ранних версиях стандартов группы Wi-Fi [5], а также (в сочетании с временным мультиплексированием) – в базовых стандартах группы GSM [1] (см. также подп. 2.2.3).

Частотное мультиплексирование, основанное на формировании сигнала-носителя данных методом модуляции с множеством взаимно ортогональных поднесущих, известно под названием *ортогонального частотного мультиплексирования* (по-англ. – *Orthogonal Frequency Division Multiplexing*, сокращенно – OFDM). В настоящее время оно широко применяется в скоростных частотно-мультиплексируемых ФКС [5]. OFDM, по сравнению с вышеописанным «классическим» FDM, обладает следующими основными преимуществами [5]:

- значительно меньшей чувствительностью к неравномерности АЧХ и фазо-частотной характеристики (ФЧХ) передающей среды в пределах диапазона частот, выделенного под ФКС;
- более эффективным использованием указанного диапазона;
- более высокой устойчивостью к *межсимвольной интерференции*, т. е. взаимному влиянию *символов* при их передаче по ФКС.

Напомним, что под символом понимается последовательность битов, представляемая в одном интервале модуляции (манипуляции) с минимальной длительностью (см. рис. 2.25, 2.26 и 2.29).

Рассмотрим вкратце основы OFDM [5].

При OFDM подлежащий передаче по ФКС поток битов демультиплексируется на N *субпотоков*. На практике N равно от нескольких десятков до нескольких тысяч. Каждый из них представляет собой формируемый по определенным правилам фрагмент исходной двоичной последовательности. Простейшим из таких правил является следующее: в i-й субпоток ($i = 1, 2, \ldots, N$) включаются биты исходной последовательности с номерами, равными $m \times N + (i - 1)$, где m – целое неотрицательное число. Например, при $N = 1024$ в 1-й субпоток включаются биты с номерами, равными $1024m$, во 2-й – $1024m + 1$ и т. д. Все субпотоки передаются параллельно во времени и независимо друг от друга, посредством модулируемых ими синусоидальных поднесущих. При этом в пределах полосы пропускания ФКС каждому

из субпотоков выделяется поднесущая определенной частоты, отличной от частот других поднесущих. Таким образом, для достижения некоторой заданной скорости передачи данных по ФКС используется N *параллельных частотных подканалов*, со скоростью передачи данных по каждому из них, в N раз меньшей общей скорости передачи по ФКС. Следовательно, ширина спектра сигнала-носителя каждого из этих подканалов может быть примерно в N раз меньше ширины спектра сигнала-носителя, которая потребовалась бы для обеспечения такой же скорости передачи при использовании одного частотного канала (см. выражения (2.7) и (2.8)).

Применение N низкоскоростных параллельных частотных подканалов, по сравнению с использованием одного скоростного, обладает следующими преимуществами.

Во-первых, оно требует равномерности АЧХ и ФЧХ передающей среды только в пределах каждого из частотных подканалов, т. е. в диапазоне частот, примерно в N раз меньшем, чем при одном скоростном канале. При этом АЧХ и ФЧХ различных подканалов могут существенно различаться между собой. Следовательно, при N низкоскоростных подканалах достоверность обмена данными значительно в меньшей степени, чем при одном скоростном, зависит от неравномерности АЧХ и ФЧХ передающей среды в пределах частотного диапазона ФКС в целом.

Во-вторых, в N раз меньшая скорость передачи данных по каждому из частотных подканалов, по сравнению с ФКС в целом, позволяет вводить защитные временные интервалы между передаваемыми по подканалам символами. Их типовая длительность составляет порядка 10 – 25 % от длительности символа. Это значительно снижает эффект межсимвольной интерференции без существенного влияния на скорость передачи.

Модуляция поднесущих частотных подканалов, как правило, реализуется методами PSK (DPSK) или QAM (см. подп. 2.6.3 и 2.6.4 соответственно). Результатом модуляции является набор N гармонических сигналов-носителей данных, модулированных по фазе (при использовании QAM – также и по амплитуде). В пределах интервала модуляции (манипуляции) с минимальной длительностью $T_{\min}$ (см. рис. 2.25, 2.26 и 2.29) начальные фазы и амплитуды этих сигналов *постоянны*.

Выбор частот поднесущих осуществляется исходя из условия их *взаимной ортогональности*, откуда и происходит термин «ортого-

нальное частотное мультиплексирование». Соблюдение данного условия позволяет с максимальной эффективностью использовать частотный диапазон, выделенный под ФКС (см. далее). Смысл условия взаимной ортогональности состоит в том, что удовлетворяющие ему поднесущие могут служить в качестве *базисных функций* (см. выражение (2.1) и пояснения к нему) при *дискретном преобразовании Фурье* (ДПФ) некоторого сигнала на интервале наблюдения с длительностью $T_{\min}$.

Математически условие взаимной ортогональности записывается следующим образом [1]:

$$\left.\begin{aligned} f_{SC1} &= k/T_{\min}; \\ \Delta f_{SC} &= m/T_{\min}; \end{aligned}\right\}, \qquad (2.32)$$

где f_{SC1} – частота первой поднесущей;

Δf_{SC} – разность между частотами соседних поднесущих;

$T_{\min}$ – минимальная длительность интервала модуляции (манипуляции) поднесущей (см. рис. 2.25, 2.26 и 2.29), т. е. длительность одного двоичного символа, представляемого модулированным сигналом;

k и m – целые положительные числа (на практике обычно $m = 1$).

При выполнении условия (2.32) набор модулированных поднесущих в пределах некоторого i-го интервала модуляции длительностью $T_{\min}$ описывается следующими выражениями:

$$\left.\begin{aligned} X_{1i}(t) &= X_{m1i} \times \cos\{2\pi \times m \times (t/T_{\min}) + \varphi_{1i}\}; \\ X_{2i}(t) &= X_{m2i} \times \cos\{2\pi \times [m+k] \times (t/T_{\min}) + \varphi_{2i}\}; \\ &\bullet \\ &\bullet \\ &\bullet \\ X_{Ni}(t) &= X_{mNi} \times \cos\{2\pi \times [m + k(N-1)] \times (t/T_{\min}) + \varphi_{Ni}\} \end{aligned}\right\}, \qquad (2.33)$$

где X_{m1i}, …, X_{mNi} и φ_{1i}, …, φ_{Ni} – соответственно амплитуды и начальные фазы сигналов-носителей данных частотных подканалов в i-м интервале модуляции длительностью $T_{\min}$.

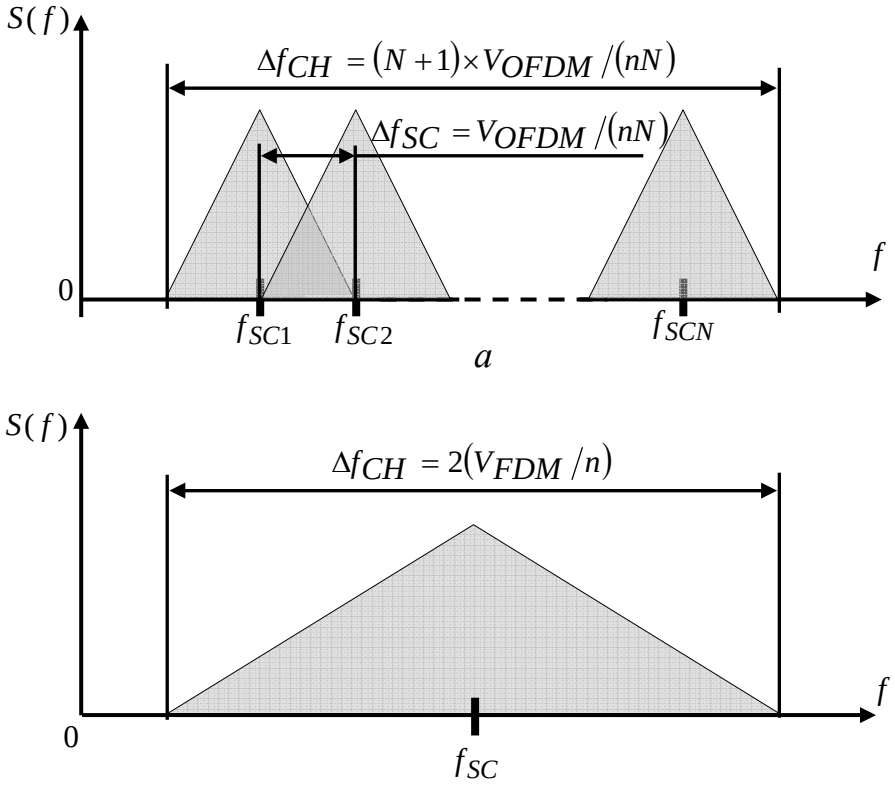
Набор сигналов (2.33), по существу, представляет собой совокупность спектральных компонент сигнала-носителя *всех N субпотоков* цифровых данных на интервале времени длительностью $T_{\min}$, т. е. является результатом дискретного преобразования Фурье (ДПФ) этого

сигнала на данном интервале. Поэтому сигнал-носитель естественно формировать как результат обратного ДПФ (ОДПФ) N модулированных поднесущих $X_{1i}(t)$, $X_{2i}(t)$, … , $X_{Ni}(t)$, что позволяет эффективно использовать диапазон частот, выделенный под ФКС. Это обусловлено допустимостью *перекрытия* частотных диапазонов, выделяемых под модулированные взаимно ортогональные поднесущие, при формировании сигнала-носителя методом их ОПДФ. При этом разность частот соседних поднесущих на практике обычно равна $1/T_{\min}$ (см. выражение (2.32) и пояснения к нему). В то же время, значимые спектральные составляющие модулированного сигнала сосредоточены в диапазоне частот шириной $2/T_{\min}$ (см. выражение (2.4), а также материалы п. 2.6). Поэтому при разности частот соседних поднесущих, равной $1/T_{\min}$, имеет место перекрытие частотных диапазонов соседних подканалов, что иллюстрирует рис. 2.43, *а*. Однако, благодаря взаимной ортогональности поднесущих, это не препятствует как корректному формированию сигнала-носителя данных методом ОДПФ на передающей стороне, так и восстановлению методом ДПФ из этого сигнала модулированных поднесущих на приемной стороне. Таким образом, при ширине выделенного под ФКС диапазона частот, равной некоторому значению Δf_{CH}, модуляция с N взаимно ортогональными поднесущими и Δf_{SC}, равном $1/T_{\min}$ (см. условия (2.32)), позволяет достигнуть значения $T_{\min}$ в одном подканале, равного $(N+1)/\Delta f_{CH}$. При этом скорость передачи данных (в битах в секунду) по одному подканалу равна приблизительно $n \times \Delta f_{CH} / (N+1)$, а по ФКС в целом – $n \times N \times \Delta f_{CH} / (N+1) \approx n \times \Delta f_{CH}$, где n – разрядность символов, передаваемых по подканалам (рис. 2.43, *а*).

С другой стороны, при использовании одной поднесущей и прочих равных условиях, а именно: такой же ширине частотного диапазона, выделенного под ФКС, одинаковом отношении «сигнал-шум» и, следовательно, той же разрядности n символа, определяемой, в первую очередь, данным отношением, достижимая длительность символа равна $2/\Delta f_{CH}$ (см. выражение (2.14) и рис. 2.43, *б*), а скорость передачи данных по ФКС – $n \times \Delta f_{CH} / 2$ бит в секунду.

Следовательно, при одинаковой ширине полосы пропускания ФКС и разрядности символа OFDM теоретически позволяет достигнуть скорости обмена данными в 2 раза большей, чем «классическое» FDM. На практике выигрыш примерно на 20 – 25 % меньше [1].

Спектр полученного в результате ОДПФ сигнала-носителя цифровых данных переносится (обычно методом амплитудной модуляции с подавленной несущей) в диапазон частот, выделенный для соответствующего ФКС. Затем сигнал передается принимающему его абоненту. На приемной стороне из сигнала-носителя методом прямого ДПФ восстанавливаются N модулированных поднесущих (см. выражение (2.33)). Каждая из них, в свою очередь, является сигналом-носителем данных одного из N частотных подканалов ФКС, представляющим определенный двоичный субпоток (см. выше). После демодуляции сигналов-носителей субпотоки объединяются (мультиплексируются) в единый поток битов.



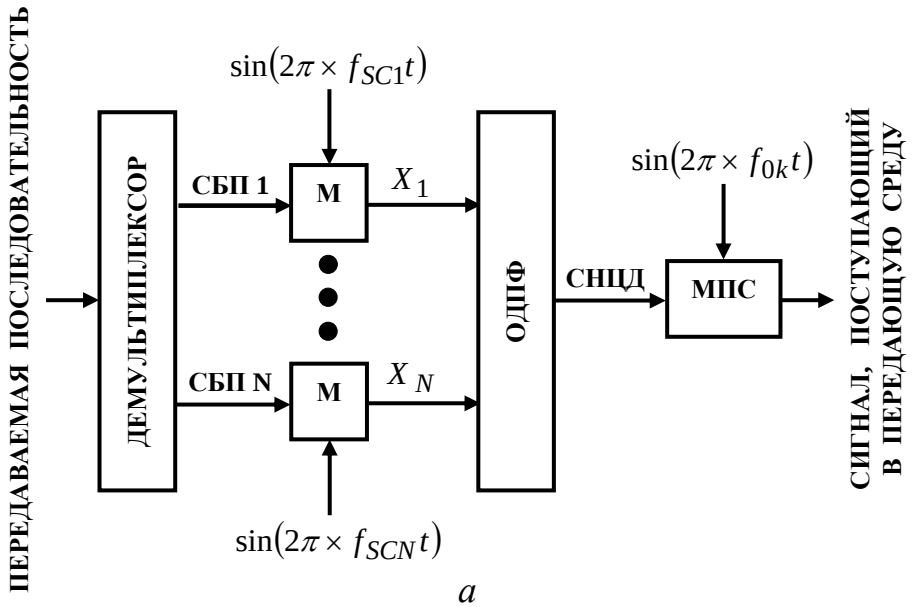
f_{SC}; $f_{SC1}, \ldots, f_{SCN}$ – частоты поднесущих;

V_{OFDM} и V_{FDM} – скорости передачи данных (в битах в секунду) при OFDM и «классическом» FDM соответственно

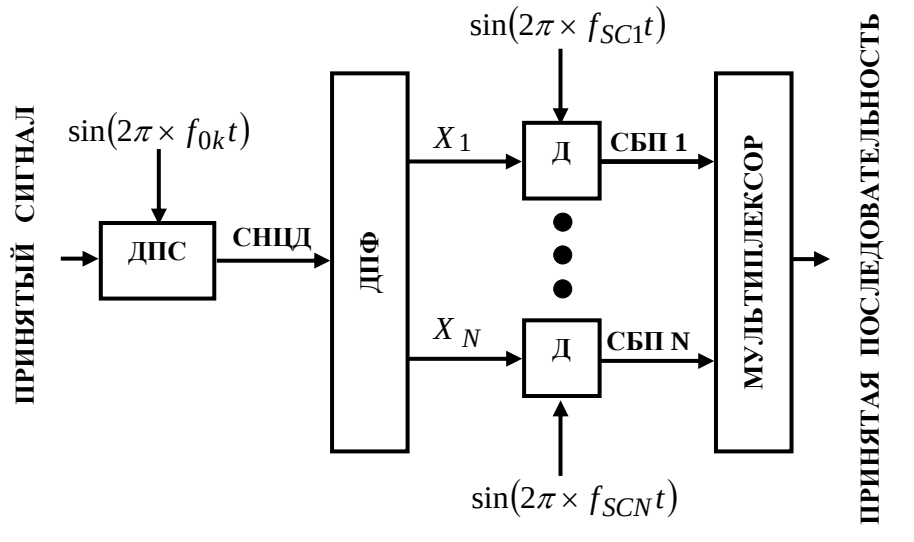
Рис. 2.43. Примеры распределения СПМ сигнала-носителя данных одного ФКС (до переноса спектра) при множестве взаимно ортогональных поднесущих (*а*) и одной поднесущей (*б*)

Упрощенная операционная модель формирования сигнала-носителя двоичных данных при OFDM представлена на рис. 2.44, *а*. На

рис. 2.44, *б* приведена также упрощенная операционная модель восстановления на приемной стороне переданных двоичных данных из указанного сигнала-носителя. Рис. 2.44, *а* и 2.44, *б* иллюстрируют изложенные ранее принципы OFDM и поэтому не нуждаются в подробных комментариях.



а



б

[3]

СБП1, …, СБПN – субпотоки двоичных данных; М – модуляторы; X_1, …, X_N – модулированные поднесущие (см. выражение (2.33)); СНЦД – сигнал-носитель N субпотоков цифровых данных; МПС – модулятор с переносом спектра; ДПС – демодулятор с переносом спектра; Д – демодуляторы; f_{SC1}, … , f_{SCN} – частоты поднесущих; f_{0k} – частота несущей, выделенная под ФКС

Рис. 2.44. Пояснения принципа реализации OFDM: упрощенная операционная модель процесса формирования сигнала-носителя цифровых данных (*а*) и их восстановления из него (*б*)

Необходимо только отметить, что на практике собственно формирование сигнала-носителя на передающей стороне и восстановление из него исходных данных на приемной обычно осуществляются в цифровой форме (аппаратной и/или программной), а перенос спектра этого сигнала при передаче и приеме – в аналоговой [5]. Последнее обусловлено высокими частотами несущих f_{0k} (до десятков ГГц в беспроводных ФКС) и, как следствие, сложностью их модуляции и демодуляции в цифровой форме.

Следует также отметить, что, в общем случае, восстанавливаемые на приемной стороне цифровые данные могут не совпадать с переданной двоичной последовательностью из-за искажений сигнала-носителя. Ошибки, вызванные этими искажениями, обнаруживаются и исправляются средствами помехоустойчивого кодирования на физическом и канальном уровне (см. рис. 2.34 и пояснения к нему).

Основным недостатком OFDM является относительно высокая сложность реализации (см. рис. 2.44) по сравнению с «классическим» FDM. Однако при современном уровне развития аппаратно-программных средств кодирования и обработки данных этот недостаток не является существенным по сравнению с вышеописанными преимуществами OFDM. Поэтому в настоящее время оно широко применяется в скоростных частотно-мультиплексируемых ФКС. В частности, OFDM используется рядом современных протоколов обмена данными на физическом уровне по беспроводным абонентским окончаниям (большинство стандартов группы WiMAX) и беспроводным ФКС ЛВС (ряд стандартов группы Wi-Fi) [5]. Также на OFDM основываются некоторые стандарты ADSL (см. подп. 2.2.1) и некоторые другие технологии группы xDSL [1, 7].

В качестве типового примера приведем сочетание базовых характеристик OFDM, оговариваемое одним из стандартов группы WiMAX [5]:

- ширина полосы частот, выделяемой под один ФКС, – 10 МГц;
- количество поднесущих (т. е. параллельных частотных подканалов) – 1024;
- ширина полосы частот, выделяемой под один частотный подканал, – 9,76 кГц;
- длительность символа, $T_{\min}$, – 102,4 мкс;
- длительность защитного интервала между символами – 12,8 мкс.

2.8.3. Волновое мультиплексирование (WDM)

Данный метод мультиплексирования, по существу, является частным случаем частотного мультиплексирования, применяемым при использовании ВОК (см. подп. 2.3.3) в качестве передающей среды. Название метода основывается на том, что для излучения инфракрасного диапазона, применяемого для представления данных в ВОК, принято указывать не частоту, а длину волны.

Принцип WDM основывается на разделении множеством ФКС одного и того же участка ВОК за счет выделения определенной длины волны для оптического сигнала-носителя данных каждого из ФКС, называемых при этом *спектральными каналами связи*. Представление данных в каждом из этих каналов осуществляется импульсами излучения с выделенной под соответствующий канал длиной волны.

По разности длин волн соседних каналов различают [3]:

- «грубое» волновое мультиплексирование (по-англ. – *Coarse Wavelength-Division Multiplexing*, сокращенно – *CWDM*), при котором указанная разность составляет 20 нм;
- «обычное» волновое мультиплексирование (по-англ. – *conventional Wavelength-Division Multiplexing*, обычно обозначаемое аббревиатурой *WDM*), при котором разность длин волн соседних каналов находится в пределах от 3,2 до 6,4 нм;
- уплотненное волновое мультиплексирование (по-англ. – *Dense Wavelength-Division Multiplexing*, сокращенно – *DWDM*), при котором указанная разность составляет порядка 0,4 – 0,8 нм;
- высокоуплотненное волновое мультиплексирование (по-англ. – *High-Dense Wavelength-Division Multiplexing*, *HDWDM*), с разностью длин волн соседних каналов порядка 0,2 нм.

Например, распространены следующие два основных варианта уплотненного волнового мультиплексирования [3]:

- 41 канал в диапазоне длин волн от 1528,77 до 1560,61 нм с разностью длин волн соседних каналов, равной приблизительно 0,8 нм;
- 81 канал в том же диапазоне длин волн, с разностью длин волн соседних каналов, равной примерно 0,4 нм.

Естественно, чем меньше разность длин волн соседних каналов, тем большее их количество может быть реализовано в пределах заданного диапазона длин волн (см. выше). С другой стороны, чем меньше указанная разность, тем сложнее, качественнее и дороже должно быть оптическое оборудование мультиплексирования и демультиплексирования WDM-системы.

Следует отметить, что в ряде практических случаев волновое мультиплексирование применяется не только с целью разделения несколькими независимыми ФКС одного и того же участка ВОК, но также для повышения скорости обмена данными по одному ФКС [1]. Действительно, если некоторая технология передачи данных по ВОК обеспечивает скорость передачи в пределах одного ФКС и на одной длине волны, равную C бит в секунду, то при использовании этим ФКС N сигналов-носителей с различной длиной волны, скорость передачи возрастает до $N \times C$ бит в секунду.

Необходимо отметить, что большинство существующих технологий волнового мультиплексирования предполагает применение одномодового оптического волокна (см. подп. 2.3.3), однако известны и WDM-системы на основе многомодового волокна.

Реализация волнового мультиплексирования осуществляется посредством специального оптического оборудования, рассмотрение принципов функционирования которого выходит за рамки настоящего пособия. Более подробно вопросы физической реализации волнового мультиплексирования изложены, например, в [3].

Волновое мультиплексирование (в основном уплотненное и высокоуплотненное) применяется, например, в протяженных, скоростных МЛС КТСОП [3, 7] (см. рис. 2.6).

2.8.4. Временное мультиплексирование (TDM)

В то время как ранее рассмотренное частотное мультиплексирование применяется как в аналоговых, так и в цифровых системах связи, временное мультиплексирование, как правило, требует цифрового представления данных. Принцип данного метода мультиплексирования достаточно прост и состоит в следующем. Каждому из ФКС, разделяющих между собой один и тот же участок передающей среды, выделяются определенные интервалы времени для обмена данными, не перекрывающиеся с интервалами времени, выделяемыми другим ФКС. Данные интервалы известны под названием *тайм-слотов* (от англ. *time-slot*, в дословном переводе – «временная щель»). Другими словами, каждый из ФКС поочередно занимает разделяемый с другими участок передающей среды на определенный интервал времени, по истечении которого передающая среда «уступается» некоторому другому ФКС.

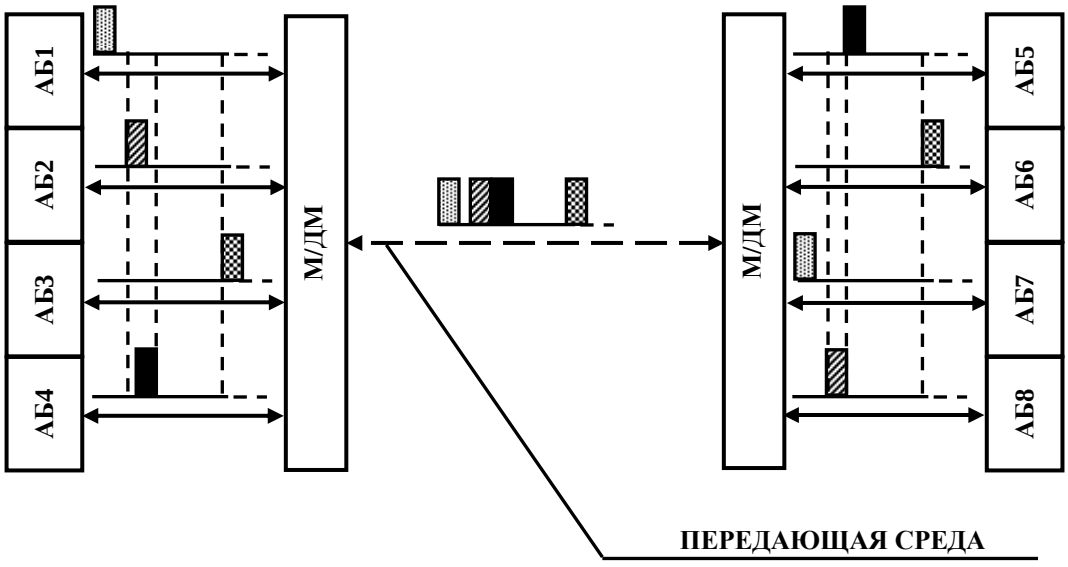
По принципам организации разделения передающей среды во времени различают два основных режима TDM: *асинхронный* и *синхронный* [3].

Асинхронный режим характеризуется отсутствием взаимной синхронизации между потоками данных различных ФКС, разделяющих между собой один и тот участок передающей среды, т. е.:

- непостоянным и изменяющимся случайным образом, в широких пределах, интервалом времени между тайм-слотами, выделяемыми различным ФКС;
- отсутствием заранее установленной очередности в доступе ФКС к передающей среде.

Другими словами, каждый ФКС занимает ее, когда, с одной стороны, в этом возникает необходимость, а с другой – передающая среда не занята другими ФКС с таким же или более высоким приоритетом доступа к ней.

Принцип асинхронного TDM иллюстрирует рис. 2.45, на котором представлена упрощенная операционная модель процесса обмена данными между 4-мя парами абонентов (т. е. по 4-м ФКС) асинхронной TDM-системы связи. При этом 1-й ФКС связывает 1-го и 7-го абонентов, 2-й – 2-го и 8-го, 3-й – 3-го и 6-го, а 4-й – 4-го и 5-го.



АБ1 – АБ8 – абоненты асинхронной TDM-системы;
М/ДМ – мультиплексоры/демультиплексоры

Рис. 2.45. Упрощенная операционная модель процесса обмена данными между абонентами асинхронной TDM-системы (см. комментарии в тексте)

Для определенности предполагается, что абоненты с 1-го по 4-й передают, а с 5-го по 8-й – принимают данные. Прямоугольниками с различной штриховкой обозначены сообщения, относящиеся к различным ФКС. Вертикальные пунктирные линии на рис. 2.45 указывают взаимное расположение этих сообщений во времени.

Мультиплексоры/демультиплексоры в режиме мультиплексирования осуществляют объединение потоков данных, поступающих от различных абонентов, в единый поток, подаваемый в передающую среду. В режиме демультиплексирования они выполняют функции выделения потоков данных, предназначенных конкретным абонентам, из общего потока данных, поступающего из передающей среды.

Необходимо отметить, что на практике функции мультиплексирования/демультиплексирования часто реализует АПД абонентов. Также следует указать, что при асинхронном TDM сообщения обычно снабжаются *адресным полем*, т. е. содержат адрес абонента – получателя. Этот адрес используется при демультиплексировании данных, предназначаемых конкретным абонентам.

Асинхронное TDM применяется, например, в технологиях ЛВС Ethernet и в сетевой технологии ATM (*Asynchronous Transfer Mode*) [3].

Синхронный режим TDM иллюстрирует рис. 2.46. Данный режим отличается от асинхронного:

- строгой очередностью следования тайм-слотов, выделяемых для различных ФКС;

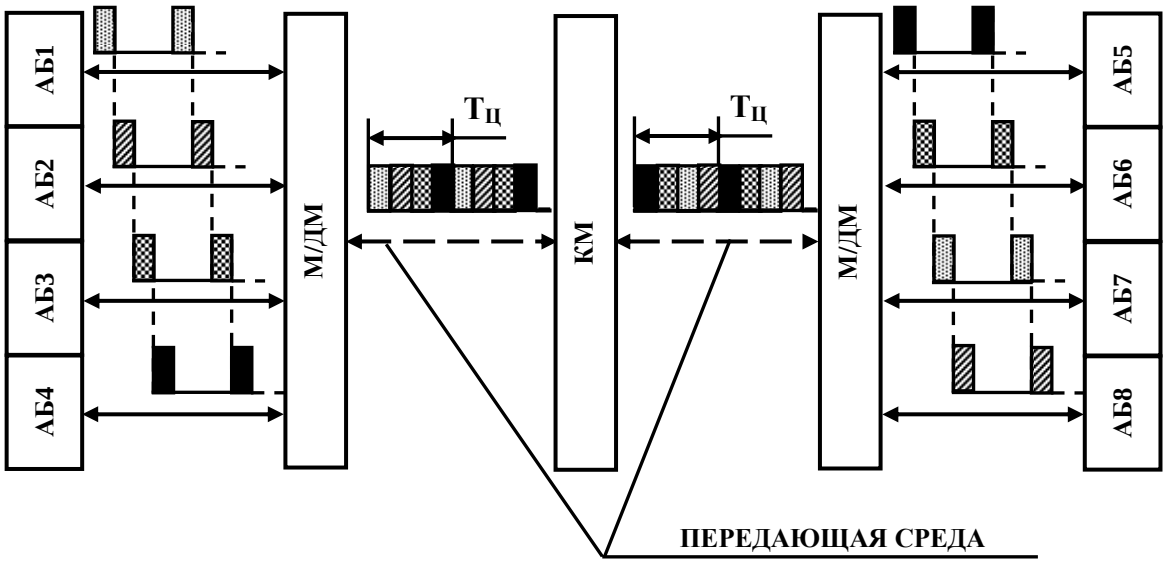
- одинаковой и практически неизменной во времени длительностью интервалов между этими тайм-слотами, причем часто эти интервалы отсутствуют.

Следует отметить, что в литературе термин «временное мультиплексирование» без конкретизации режима обычно по умолчанию предполагает синхронный режим TDM [3].

Предоставление доступа ФКС к передающей среде в синхронном режиме обычно осуществляется *циклически*: каждый из ФКС по очереди занимает ее на определенный интервал времени, по истечении которого передающая среда предоставляется следующему по порядку ФКС. По окончании интервала, выделенного для последнего по очереди ФКС, доступ к передающей среде снова переходит к первому ФКС и так далее. Интервал времени, в течение которого каждому из ФКС, разделяющих один и тот же участок передающей среды, предоставляется по одному тайм-слоту для доступа к ней, называется *циклом обмена данными* или просто *циклом*.

Принцип синхронного временного мультиплексирования иллюстрирует рис. 2.46, на котором представлена упрощенная операционная модель процесс обмена данными между 4-мя парами абонентов (т. е. по 4-м ФКС) синхронной TDM-системы связи. Рис. 2.46, как и 2.45 предполагает, что 1-й ФКС связывает 1-го и 7-го абонентов, 2-й – 2-го и 8-го, 3-й – 3-го и 6-го, а 4-й – 4-го и 5-го, причем абоненты с 1-го по 4-й передают, а с 5-го по 8-й – принимают данные. Условные обозначения рис. 2.46 аналогичны приведенным на рис. 2.45.

Необходимо отметить, что в синхронном режиме, в отличие от асинхронного, сообщения обычно не содержат адресной информации, а демультиплексирование на приемной стороне (т. е. выделение данных, предназначенных конкретным абонентам) осуществляется в зависимости от времени поступления сообщения. Например, сообщение, начало которого отстоит на $i-1$ тайм-слотов от начала цикла обмена данными, поступает i-му принимающему абоненту (рис. 2.46).



АБ1 – АБ8 – абоненты синхронной TDM-системы;
М/ДМ – мультиплексоры/демультиплексоры; КМ – коммутатор;
$T_Ц$ – длительность цикла обмена данными

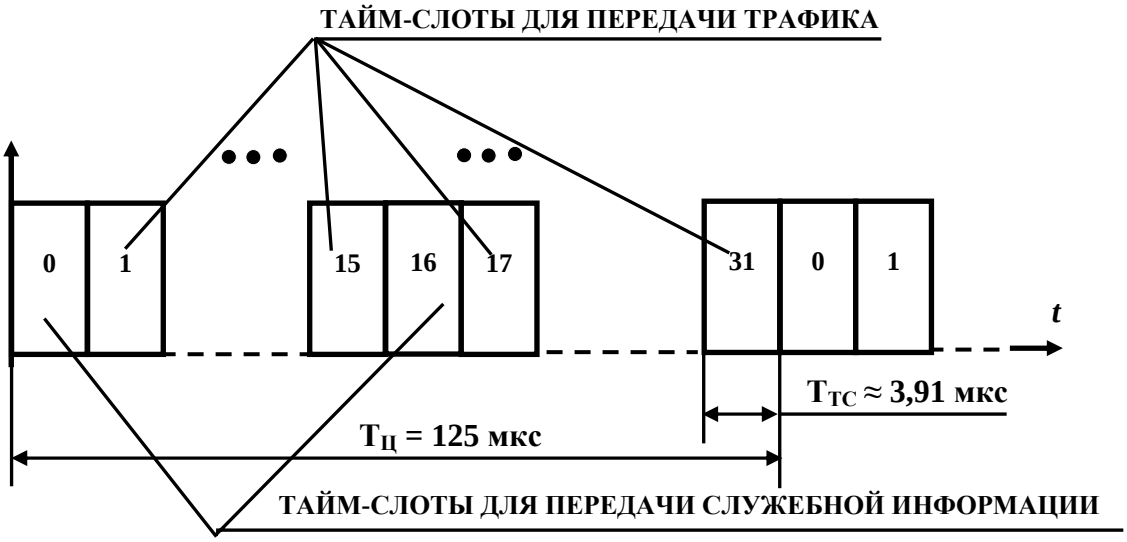
Рис. 2.46. Упрощенная операционная модель процесса обмена данными между абонентами синхронной TDM-системы (см. комментарии в тексте)

Указанный способ демультиплексирования требует наличия:

- быстродействующего коммутатора с памятью в составе TDM-системы, который осуществляет перераспределение сообщений в общем потоке в соответствии с порядком их подачи принимающим абонентам (см. рис. 2.46);

- *цикловых синхросигналов* (*синхросимволов*) в структуре цикла, по которым определяется его начало при демультиплексировании (на рис. 2.46 не показаны; см. приведенный на рис. 2.47 пример цикла обмена данными).

Простейший практический пример цикла обмена данными приведен на рис. 2.47. На нем изображена структура такого цикла в *канале связи типа Е-1* [3].



$T_Ц$ – длительность цикла обмена данными; $T_{ТС}$ – длительность тайм-слота

Рис. 2.47. Структура цикла обмена данными в канале типа Е-1

Он представляет собой совокупность 32-х мультиплексированных во времени ФКС, разделяющих один и тот же участок передающей среды и предназначенных для обмена данными в цифровой форме между концентраторами и опорными станциями КТСОП (см. рис. 2.6 и пояснения к нему). Два из этих ФКС (0-й и 16-й) используются для передачи служебной информации, в том числе синхросимволов (см. выше), а остальные – для формирования трафика, представляющего собой 30 разделенных во времени, независимых потоков данных соответственно между 30-ю парами абонентов. В их качестве могут выступать как телефонные аппараты или факсы, так и АПД абонентских компьютеров ВС.

Каждому из ФКС канала Е-1 поочередно выделяется тайм-слот (называемый также *канальным интервалом* или *импульсным каналом*) длительностью примерно 3,91 мкс. В течение этого тайм-слота по соответствующему ФКС передается один байт данных, представляемый

линейным кодом HDB3 (см. рис. 2.21 и пояснения к нему). В качестве данных служат цифровые отсчеты информативных сигналов абонентских линий связи КТСОП (см. рис. 2.6). Напомним, что эти сигналы находятся в частотном диапазоне от 300 до 3400 Гц. Поэтому частота их дискретизации в соответствии с теоремой Найквиста должна быть больше 6800 Гц; на практике в каналах связи Е-1 она равна 8000 Гц. Такова же частота повторения тайм-слотов, выделяемых одному и тому же ФКС. При этом длительность цикла обмена данными, очевидно, равна 1/8000 Гц = 125 мкс (см. рис. 2.47). Скорость передачи данных по одному ФКС составляет 64 Кбит в сек., а по каналу Е-1 в целом – 2048 Кбит в сек.

Необходимо отметить, что во многих синхронных TDM-системах, в том числе в коммутационных системах КТСОП, применяется *многоуровневое* временное мультиплексирование [3]. Так, например, путем объединения методом TDM 4-х каналов Е-1 формируется *канал связи типа Е-2*, представляющий собой совокупность 120-ти разделенных во времени, независимых абонентских ФКС, разделяющих между собой один и тот же участок передающей среды КТСОП. В свою очередь, 4 канала Е-2 также методом TDM объединяются в канал связи типа Е-3 (480 разделенных во времени абонентских ФКС), а 4 канала Е-3 – в канал типа Е-4 (1920 мультиплексированных во времени пользовательских ФКС). Наконец, 4 канала Е-4 объединяются в канал связи типа Е-5 (7680 абонентских ФКС, разделяющих методом TDM один и тот же участок передающей среды КТСОП). Описанная система многоуровневого TDM соответствует *PDH-системе Е-каналов* (от англ. словосочетания *Plesiochronous Digital Hierarchy*, в переводе – *плезиохронная цифровая иерархия*). Данная система применяется в КТСОП европейских стран, в том числе России. Существуют также PDH-системы Т-каналов (страны Северной Америки) и DS-каналов (Япония) [3]. Термин «плезиохронная» (в дословном переводе – «почти синхронная») означает отсутствие полной взаимной синхронности потоков данных при объединении каналов более низкого уровня иерархии в канал более высокого уровня (см. выше), в связи с чем при таком объединении принимаются специальные меры [3]. Следует отметить, что PDH-система является первым поколением применяемых в КТСОП TDM-технологий. В настоящее время она применяется, в основном, в «периферийных» МЛС КТСОП. В протяженных, скоростных МЛС используется более современная SDH-система каналов связи с временным мультиплекси-

рованием (от англ. словосочетания *Synchronous Digital Hierarchy*, в переводе – *синхронная цифровая иерархия*) [3]. Следует, однако, отметить, что в настоящее время SDH-технология, в свою очередь, постепенно вытесняется уплотненным и высокоуплотненным волновым мультиплексированием с основных магистралей КТСОП на их периферию [3].

Кроме МЛС КТСОП (см. рис. 2.6), применение синхронного временного мультиплексирования (в сочетании с частотным мультиплексированием) также оговаривается рядом версий стандартов мобильной связи группы GSM [5] (см. подп. 2.2.3). Так, согласно одной из них, под каждый из физических каналов передачи данных, как от МС к БППС, так и в обратном направлении, отводится некоторый частотный диапазон шириной 200 кГц в пределах полосы частот 890,2 – 914,8 МГц или 935,2 – 959,8 МГц соответственно. В свою очередь, каждый из вышеуказанных диапазонов частот шириной 200 кГц используется для связи с БППС одновременно 8 МС, каждой из которых поочередно выделяется интервал времени (называемый «*тайм-слот*») длительностью 577 мкс для обмена данными с БППС.

2.8.5. Кодовое мультиплексирование (CDMA)

Данный метод мультиплексирования является сравнительно новым. Его общий принцип состоит в следующем. Ряд ФКС (на практике – несколько десятков [3, 7]), разделяющих между собой некоторый участок передающей среды, используют для обмена данными один и тот же частотный диапазон и одни и те же интервалы времени. Исключение взаимного влияния потоков данных, передаваемых по различным ФКС, осуществляется за счет представления каждого из этих потоков некоторым *индивидуальным кодом*, не совпадающим с кодами, применяемыми в других ФКС. Образно говоря, в каждом из ФКС используется свой «язык общения» между абонентами, отличающийся от «языка», применяемого в других ФКС. Поэтому абоненты каждого из ФКС воспринимают только информацию на «своем языке» (т. е. представленную «своим» кодом), игнорируя сообщения на «чужих языках».

Рассмотрим вкратце принцип кодового мультиплексирования ФКС ВС [3, 7].

Для кодирования и декодирования двоичных данных при CDMA характерна их биполярная интерпретация: логическая единица пред-

ставляется как +1, а ноль – как -1. Кодирование передаваемых данных осуществляется путем побитового алгебраического перемножения двух представленных в биполярной форме двоичных последовательностей: подлежащей передаче и *идентифицирующей последовательности* (ИДП), называемой также *элементарной последовательностью*, которая назначается индивидуально для каждого из ФКС и отлична от ИДП всех других ФКС, совместно использующих передающую среду. ИДП CDMA-системы связи обладают следующими основными свойствами:

- длительность их битовых интервалов в m раз меньше, чем у исходной последовательности, подлежащей передаче, где m – целое число, оговариваемое конкретным протоколом CDMA и обычно равное 64-м или 128-ми [1]; при этом биты ИДП часто называют *чипами* (*chips*) или *элементарными сигналами*;

- ИДП различных ФКС *взаимно ортогональны*, т. е. удовлетворяют следующим условиям:

$$\left.\begin{aligned} \frac{1}{m}\sum_{i=0}^{m-1} ID_{ij}\,ID_{ik}\Big|_{j\neq k} &= 0;\\ \frac{1}{m}\sum_{i=0}^{m-1} ID_{ij}\,ID_{ik}\Big|_{j=k} &= 1;\\ \frac{1}{m}\sum_{i=0}^{m-1} ID_{ij}\left(-ID_{ik}\right)\Big|_{j\neq k} &= 0;\\ \frac{1}{m}\sum_{i=0}^{m-1} ID_{ij}\left(-ID_{ik}\right)\Big|_{j=k} &= -1; \end{aligned}\right\}, \qquad (2.34)$$

где ID_{ij} и ID_{ik} – состояние (которое может быть равно +1 или -1) i-го разряда ИДП соответственно j-го и k-го ФКС.

Перечисленные свойства ИДП позволяют абонентам каждой из ФКС обнаруживать (идентифицировать) предназначенные им сообщения в совокупности сигналов, поступающих из разделяемой с другими ФКС передающей среды (см. далее).

Известны два основных типа ИДП: *детерминированные* и *псевдослучайные*.

Детерминированная ИДП представляет собой m-разрядную биполярную двоичную последовательность, назначенную соответствующему ФКС, длительность которой совпадает с длительностью битового интервала передаваемого сообщения. Его кодирование детерминированной ИДП состоит в представлении ею каждой из единиц сооб-

щения, а каждого из нулей – инверсией этой ИДП. При этом под инверсией биполярной двоичной последовательности понимается замена знака каждого из ее битов. Применение детерминированной ИДП требует синхронизации начал битовых интервалов исходной последовательности с началами ИДП как на передающей, так и на приемной стороне. Поэтому CDMA-системы, в которых они применяются, называются *синхронными*. Практически синхронизация осуществляется путем введения в состав ФКС CDMA-системы специальных каналов для передачи синхронизирующей информации. В качестве ИДП синхронных CDMA-систем выступают двухуровневые биполярные функции, принадлежащие к некоторой *ортогональной системе*. Часто в их качестве применяются *функции Уолша* [5]. Пример ортогональной системы двухуровневых функций будет представлен далее (см. выражения (2.36)). Основным недостатком CDMA-систем связи, использующих детерминированные ИДП, является сложность обеспечения синхронизации мобильных абонентов.

Псевдослучайная ИДП представляет собой биполярную двоичную псевдослучайную последовательность, *некоррелированную* (т. е., упрощенно говоря, взаимно независимую) с ИДП других ФКС, использующих тот же участок передающей среды. Длительность битового интервала этих ИДП, как и детерминированных, в m раз меньше, чем у исходной последовательности. Однако в отличие от детерминированных, псевдослучайные ИДП не обладают повторяемостью от одного битового интервала исходного сообщения к другому. Некоррелированный характер псевдослучайных ИДП обеспечивает выполнение условий (2.34). Генерация таких ИДП осуществляется посредством реализуемых аппаратно или программно ГПП (см. подп. 2.7.3) с индивидуально назначаемыми каждому из ФКС образующими полиномами (которые, естественно, одинаковы для обоих абонентов ФКС). Применение псевдослучайных ИДП не требует взаимной синхронизации передающего и принимающего абонентов, поэтому такие ИДП используются в *асинхронных* CDMA-системах. В частности, их применение при мобильном характере абонентов более предпочтительно [1]. Основными недостатками CDMA-систем, использующих псевдослучайные ИДП, являются меньшая достоверность восстановления данных на приемной стороне и более жесткие требования к равенству амплитуд сигналов-носителей ФКС, чем при детерминированных ИДП. Однако эти недостатки преодолеваются за счет соответственно помехоустойчивого кодирования передаваемых данных и автоматического регулирования амплитуд сигналов-носителей.

Двоичная биполярная последовательность, полученная путем перемножения подлежащего передаче кода и ИДП соответствующего ФКС, затем преобразуется в модулированный синусоидальный сигнал и поступает в передающую среду. При этом применяются методы модуляции, позволяющие с помощью ИДП выделять сообщения, адресованные каждому из ФКС, из результата детектирования суммы сигналов-носителей данных всех ФКС, совместно использующих передающую среду. Это имеет место, например, если биполярная двоичная последовательность, полученная в результате детектирования указанной суммы, равна *поэлементной сумме с ограничениями по уровням ±1* биполярных последовательностей, представляемых сигналами-носителями. Данная сумма вычисляется по следующим правилам: ее значение принимается равным плюс единице, если алгебраическая сумма слагаемых больше или равна +1; минус единице, если эта сумма меньше или равна -1 и нулю – при равенстве нулю алгебраической суммы слагаемых. Указанным свойством обладает, например, результат детектирования суммы DPSK-сигналов (см. подп. 2.6.3), приблизительно равных по амплитуде. Поэтому DPSK (с автоматическим выравниванием амплитуд сигналов-носителей ФКС) достаточно широко применяется в CDMA-системах.

Абоненты всех ФКС постоянно подвергают демодуляции сумму сигналов-носителей, поступающих из передающей среды, совместно используемой этими ФКС. Результатом демодуляции является побитовая сумма с ограничениями по уровням ±1 (см. выше) представляемых этими сигналами биполярных двоичных последовательностей. Каждая из них, в свою очередь, является произведением двоичного сообщения, передаваемого одной из активных в текущий момент времени ФКС, на ИДП этой ФКС. Указанный результат демодуляции непрерывно проверяется абонентами всех ФКС на наличие в нем предназначенных им сообщений. Проверка осуществляется путем его декодирования идентифицирующей последовательностью ФКС, к которой принадлежит конкретный абонент. Декодирование осуществляется в соответствии со следующим выражением:

$$DC_j[k] = \sum_{i=0}^{m-1} DM_i[k] \times ID_{ij}[k], \qquad (2.35)$$

где $DC_j[k]$ – k-й разряд результата декодирования демодулированной последовательности посредством ИДП j-го ФКС;

$DM_i[k]$ – i-й бит результата демодуляции на k-м битовом интервале результата декодирования (напомним, что его длительность совпадает с таковой битового интервала исходного сообщения и в m раз больше, чем длительность битовых интервалов результата демодуляции и ИДП);

$ID_{ij}[k]$ – i-й бит ИДП j-го ФКС на k-м битовом интервале результата декодирования (причем у детерминированных ИДП $ID_{ij}[k]$ не зависит от k).

Суммирование в выражении (2.35) осуществляется *с ограничениями по уровням ±1*; $DM_i[k]$ и $DC_j[k]$ могут принимать значения +1, 0 или -1, а $ID_{ij}[k]$ – +1 или -1 (см. выше).

Если сообщения, адресованные абонентам j-го ФКС, отсутствуют в передающей среде на k-м битовом интервале, значение $DC_j[k]$ теоретически будет равно нулю благодаря взаимной ортогональности ИДП различных ФКС (см. выражения (2.34)). На практике это значение по абсолютной величине ниже некоторого порога детектирования. При наличии в передающей среде сообщения, адресованного соответствующему абоненту, т. е. закодированного «его» ИДП, согласно выражениям (2.34), $DC_j[k]$ в идеале будет равен плюс единице в течение единичных битовых интервалов указанного сообщения и минус единице – в течение нулевых (представляемых, как указано ранее, значением -1). На практике указанные результаты соответственно выше некоторого положительного и ниже некоторого отрицательного порога детектирования. Таким образом, ИДП позволяют не только обнаруживать, но и декодировать сообщения, адресованные конкретным абонентам.

Естественно, все вышеизложенное имеет место при отсутствии в передающей среде шумов и помех, искажающих данные, получаемые на приемной стороне. При наличии таких шумов и помех декодируемые сообщения будут содержать ошибки, устраняемые средствами помехоустойчивого кодирования (см. рис. 2.34 и пояснения к нему).

Поясним принцип CDMA простым примером. Пусть имеется синхронная CDMA-система, включающая в себя 4 независимых ФКС (на практике, естественно, число ФКС такой системы намного больше). Каждому из них присвоена детерминированная ИДП, разрядность которой для простоты положим равной 8-ми (на практике, как указано

ранее, разрядность ИДП обычно составляет 64 или 128). Условиям взаимной ортогональности (2.34) удовлетворяют, например, следующие 8-разрядные ИДП:

$$\left.\begin{array}{l} ID_0 = (-1 \quad -1 \quad -1 \quad +1 \quad +1 \quad -1 \quad +1 \quad +1); \\ ID_1 = (-1 \quad -1 \quad +1 \quad -1 \quad +1 \quad +1 \quad +1 \quad -1); \\ ID_2 = (-1 \quad +1 \quad -1 \quad +1 \quad +1 \quad +1 \quad -1 \quad -1); \\ ID_3 = (-1 \quad +1 \quad -1 \quad -1 \quad -1 \quad -1 \quad +1 \quad -1) \end{array}\right\}, \qquad (2.36)$$

где ID_j – ИДП j-го ФКС.

Пусть, например, по 0-му ФКС передается двоичный код $(1 \quad 0 \quad 1)$, а по 2-му – $(0 \quad 1 \quad 1)$, представляемые при CDMA последовательностями $(+1 \quad -1 \quad +1)$ и $(-1 \quad +1 \quad +1)$ соответственно. По 1-му и 3-му ФКС при этом данные не передаются. Тогда после умножения каждого из разрядов последовательности, подлежащей передаче по 0-му ФКС, на его ИДП ID_0 (см. выражения (2.36)), получаем следующую биполярную двоичную последовательность:

(-1 -1 -1 +1 +1 -1 +1 +1 +1 +1 +1 -1 -1 +1 -1 -1 -1 -1 -1 +1 +1 -1 +1 +1). (2.37)

С другой стороны, результат умножения каждого из битов сообщения, передаваемого по 2-му ФКС, на его ИДП ID_2 (см. выражения (2.36)), имеет следующий вид:

(+1 -1 +1 -1 -1 -1 +1 +1 -1 +1 -1 +1 +1 +1 -1 -1 -1 +1 -1 +1 +1 +1 -1 -1). (2.38)

При отсутствии в передающей среде шумов и помех, искажающих результат демодуляции, он будет равен поэлементной сумме последовательностей (2.37) и (2.38) с ограничениями по уровням ±1, представляющей собой последовательность вида:

(0 -1 0 0 0 -1 +1 +1 0 +1 0 0 0 +1 -1 -1 -1 0 -1 +1 +1 0 0 0). (2.39)

При декодировании младших 8-ми разрядов результата демодуляции (2.39) идентифицирующей последовательностью 0-го ФКС ID_0 (см. (2.36)), в соответствии с выражением (2.35) получаем, что 0-й бит декодированного сообщения равен:

(-1)×(-1) + 0×(-1) + (-1) ×(-1) + 1×1 + 1×1 + 0×(-1) + 0×1 + 0×1
(с ограничением по уровням ±1) = +1.

Декодируя аналогичным образом последовательностью ID_0 остальные две 8-разрядные группы результата демодуляции (2.39), в целом получаем сообщение $(+1 \quad -1 \quad +1)$. С другой стороны, результатом декодирования кодовой комбинации (2.39) идентифицирующей последовательностью 2-го ФКС ID_2 будет сообщение $(-1 \quad +1 \quad +1)$; а идентифицирующими последовательностями 1-го и 3-го ФКС, ID_1 и ID_3 – $(0 \quad 0 \quad 0)$. Таким образом, принимающие абоненты 0-го и 2-го ФКС, в результате декодирования «своими» ИДП демодулированной последовательности, восстановят переданные им сообщения, а 1-го и 3-го ФКС – выявят, что в данный момент в передающей среде нет адресованных им сообщений.

Следует отметить, что представленное выше описание CDMA является достаточно упрощенным. Более подробное изложение принципов CDMA, а также описание распространенных вариантов его практической реализации представлено в специальной литературе.

Основными преимуществами CDMA по сравнению с FDM и TDM являются:

- более высокая устойчивость к узкополосным помехам, чем при FDM;

- при соблюдении ряда условий (см. далее) – более высокая скорость обмена данными, чем при FDM и TDM.

Первое из перечисленных преимуществ обусловлено значительно большей шириной частотного диапазона сигнала-носителя данных каждого из ФКС CDMA-системы, чем FDM-системы с таким же количеством ФКС. В самом деле, если M ФКС CDMA-системы совместно используют некоторый диапазон частот шириной Δf, под сигнал-носитель данных каждого из этих ФКС выделяется весь указанный частотный диапазон. В то же время в FDM-системе с таким же количеством ФКС и суммарной шириной выделенного для них диапазона частот, равной Δf, под сигнал-носитель каждого из них будет выделен частотный диапазон шириной $\Delta f / M$. На практике значение M равно минимум нескольким десяткам [3, 7]. Поэтому в CDMA-системе помеха с шириной спектра $\Delta f / M$ вызовет только относительно небольшое повышение BER ФКС (см. подп. 2.2.3), так как значение $\Delta f / M$ составляет всего несколько процентов от ширины Δf частотного диапазона сигнала-носителя данных каждого из ФКС. Указанное повышение BER достаточно просто устраняется

средствами помехоустойчивого кодирования (см. рис. 2.34 и пояснения к нему). С другой стороны, помеха с шириной спектра $\Delta f / M$ вызовет практически полное нарушение процесса обмена данными, по крайней мере, по одному из ФКС FDM-системы.

Как следует из вышесказанного, CDMA является одной из технологий *расширения спектра* сигнала-носителя данных ФКС. Смысл, общие вопросы и основные методы расширения спектра будут освещены далее, в п. 2.9.

Второе из перечисленных преимуществ CDMA, увеличение скорости обмена данными по сравнению с FDM и с TDM, достигается только при определенных условиях, что нуждается в специальных пояснениях. С одной стороны, каждый из M ФКС CDMA-системы (см. выше), при прочих равных условиях, использует для связи частотный диапазон в M раз шире, чем ФКС FDM-системы и интервал времени с длительностью, в M раз большей, чем ФКС TDM-системы. Указанное должно было бы обеспечить скорость обмена данными, в M раз большую, чем при FDM и TDM. Однако вследствие кодирования передаваемых сообщений идентифицирующей последовательностью с длительностью битового интервала в m раз меньшей, чем у этих сообщений, спектр сигнала-носителя ФКС CDMA-системы, при прочих равных условиях, в m раз шире, чем FDM- и TDM-систем (см. выражение (2.4)). Это обусловливает снижение скорости обмена данными в m раз по сравнению с отсутствием кодирования идентифицирующей последовательностью. Поэтому выигрыш в скорости передачи данных при CDMA по сравнению с FDM и TDM имеет место только при M, большем m.

Кроме вышеперечисленных, CDMA обладает рядом дополнительных преимуществ, важных при применении в системах мобильной связи, в том числе в ВС с мобильными абонентами [3, 7].

Основным недостатком CDMA по сравнению с FDM и с TDM является относительно высокая сложность реализации. Однако в настоящее время этот недостаток не является существенным. Поэтому CDMA находит все более широкое распространение в скоростных протоколах обмена данными по ФКС ВС, особенно беспроводным [3, 7].

Необходимо также отметить, что на практике достаточно распространена комбинация CDMA с другими методами мультиплексирования, в частности, с TDM (сочетание CDMA с TDM известно под

названием TD-CDMA [7]). Несмотря на кажущуюся нерациональность такой комбинации, она позволяет достигнуть компромисса между числом ФКС, совместно использующих передающую среду, и качеством связи (см. далее). Примером TD-CDMA является система широкополосного TD-CDMA, TD-WCDMA. Один из типовых протоколов TD-WCDMA, UMTS (Universal Mobile Telecommunications Serv1ices) оговаривает следующие базовые характеристики TD-CDMA-системы [7]:

- ширина диапазона частот, отводимого под один частотный канал – 5 МГц;
- число тайм-слотов, выделяемых для связи между абонентами в пределах частотного канала – 16;
- число ФКС, одновременно использующих один и тот же тайм-слот одного и того же частотного канала – 64.

Нетрудно увидеть, что общее число ФКС, совместно разделяющих диапазон частот шириной 5 МГц, при этом равно 1024. В отсутствие мультиплексирования во времени потребовалось бы кодовое мультиплексирование 1024-х ФКС в пределах одного и того же частотного диапазона и интервала времени, что не позволило бы обеспечить приемлемую достоверность декодирования данных приемниками ФКС.

2.8.6. Техническая реализация мультиплексирования / демультиплексирования

Данная реализация осуществляется посредством того же сетевого оборудования, что и реализация других функций физического уровня модели OSI (см. подп. 2.5.7, 2.6.6 и 2.7.5). Принципы построения этого оборудования будут рассмотрены в гл. 3.

Выводы по п. 2.8

Основными методами мультиплексирования, применяемыми в ФКС ВС, являются частотное (FDM), волновое (WDM), временное (TDM) и кодовое (CDMA), а также их комбинации.

Частотное мультиплексирование (FDM) является исторически наиболее ранним методом мультиплексирования, применимым при обмене данными и в аналоговой, и в цифровой форме. В настоящее

время он практически не используется в проводных ФКС, включая ФКС КТСОП. Также «классическое» FDM в современных беспроводных ФКС ВС применяется только в сочетании с другими методами мультиплексирования, например, с TDM или/и с CDMA [1, 3, 7]. Более эффективное OFDM (см. подп. 2.8.2) относительно широко используется в беспроводных ФКС ВС. В частности, как указано ранее, оно применяется рядом протоколов обмена данными на физическом уровне по беспроводным абонентским окончаниям (большинство стандартов группы WiMAX) и беспроводным ФКС ЛВС (ряд стандартов группы Wi-Fi) [5]. Также на OFDM основываются некоторые стандарты ADSL (см. подп. 2.2.1) и других технологий группы xDSL [1, 7].

Волновое мультиплексирование (WDM), будучи частным случаем FDM для сигналов-носителей оптического диапазона, используется при обмене цифровыми данными в ФКС на основе ВОК. Многие протоколы обмена данными на физическом уровне по ВОК предусматривают применение WDM совместно с TDM [3].

Временное мультиплексирование (TDM) широко применяется при обмене данными в цифровой форме как в кабельных, так и в беспроводных ФКС, в том числе в сочетании с другими методами мультиплексирования. Например, на TDM, наряду с WDM (в том числе в сочетании с ним) основывается формирование потоков данных в МЛС КТСОП [3, 7]. Кроме того, TDM, совместно с CDMA или/и с FDM, достаточно широко используется в мобильной телефонной связи [5].

Кодовое мультиплексирование (CDMA) является наиболее сложным в реализации методом из вышеперечисленных, применимым только при обмене данными в цифровой форме. В то же время потенциально он наиболее эффективен с точки зрения использования выделенных под мультиплексируемые каналы связи диапазона частот и интервалов времени [3, 7]. Поэтому CDMA находит все более широкое применение при мультиплексировании ФКС, преимущественно – беспроводных [7]. Как правило, CDMA используется в сочетании с FDM и TDM (см. пример, представленный в подп. 2.8.5).

В целом, ни один из рассмотренных в данном параграфе методов мультиплексирования не является универсальным. Степень предпочтительности того или иного метода определяется характеристиками, параметрами и конкретными особенностями той или иной системы мультиплексируемых ФКС.

2.9. Расширение спектра сигналов-носителей данных ФКС ВС

2.9.1. Общие положения

Расширение спектра сигнала-носителя данных является одним из способов повышения эффективности обмена данными по ФКС, преимущественно – беспроводным. Физическая сущность расширения спектра состоит в преднамеренном многократном расширении диапазона частот, в котором находятся спектральные составляющие сигнала-носителя. Такое преобразование спектра, на первый взгляд, нерационально с точки зрения пропускной способности ФКС (см. выражения (2.8) и (2.13)). Однако, с другой стороны, расширение спектра обеспечивает [3]:

- повышение устойчивости обмена данными к узкополосным помехам, в том числе преднамеренно генерируемым «злоумышленником» с целью нарушить обмен данными по ФКС;

- снижение плотности потока излучения в ФКС, т. е. мощности излучения на единицу частоты (что важно, например, в спутниковых системах связи);

- защиту передаваемых по ФКС данных от несанкционированного перехвата узкополосным приемником, так как при этом на его вход будет поступать только незначительная часть данных, восстановление из которой сообщения в целом практически невозможно;

- возможность использования диапазона частот, выделенного под ФКС с расширенным спектром сигнала-носителя, также несколькими узкополосными ФКС, при их минимальном влиянии на широкополосный ФКС (и наоборот).

Указанные преимущества технологий расширения спектра особенно актуальны при обмене данными по беспроводным ФКС. Поэтому расширение спектра применяется в основном в данной разновидности ФКС. При этом компенсация проигрыша в пропускной способности, вызванного расширением спектра, осуществляется явным или неявным расширением диапазона частот, выделенного под некоторый ФКС или группу мультиплексируемых ФКС (что значительно проще реализуемо технически и организационно в беспроводных ФКС, чем в кабельных).

Известны следующие основные методы расширения спектра сигналов-носителей данных в беспроводных ФКС ВС [3, 5]:

- метод скачкообразной перестройки частоты (по-англ. – Frequency Hopping Spread Spectrum, *FHSS*);
- метод прямой последовательности (по-англ. – Direct Sequence Spread Spectrum, *DSSS*);
- метод включения/выключения несущей по псевдослучайному закону (по-англ. – Time Hopping Spread Spectrum, *THSS*);
- метод линейной частотной модуляции (по-англ. – Chirp Spread Spectrum, *CSS*).

Рассмотрим вкратце принципы реализации и основные свойства каждого из перечисленных методов.

2.9.2. Расширение спектра методом FHSS

Данный метод предполагает применение модулированного сигнала-носителя данных ФКС и состоит в периодическом скачкообразном изменении частоты несущей указанного сигнала. Изменение осуществляется в соответствии с некоторой закономерностью, «известной» как передатчику, так и приемнику, как правило – по псевдослучайному закону (однако, известны и варианты реализации FHSS с детерминированным законом изменения частоты) [3, 5]. Один из простейших вариантов FHSS описывается следующим выражением:

$$f_0[iT] = f_{0R} + \Delta f \times PRG[iT], \qquad (2.40)$$

где $f_0[iT]$ – частота несущей в i-м такте;

T – длительность такта переключения частоты;

f_{0R} – опорное значение частоты несущей;

$PRG[iT]$ – значение выхода генератора псевдослучайных чисел (ГПЧ) в i-м такте.

При этом ГПЧ реализуется (аппаратно или программно) аналогично ГПП (см. рис. 2.32), за исключением того, что в качестве выхода ГПЧ, в отличие от ГПП, служит $n+1$-битовое содержимое сдвигового регистра в целом. Естественно, образующие полиномы (см. выражение (2.23)) и начальные состояния ГПЧ передатчика и приемника должны быть идентичны, что обеспечивается протоколами связи и специальными процедурами установки ГПЧ в одинаковые начальные состояния на этапе инициализации сеанса связи.

Пример перестройки частоты несущей при FHSS в соответствии с выражением (2.40) приведен на рис. 2.48.

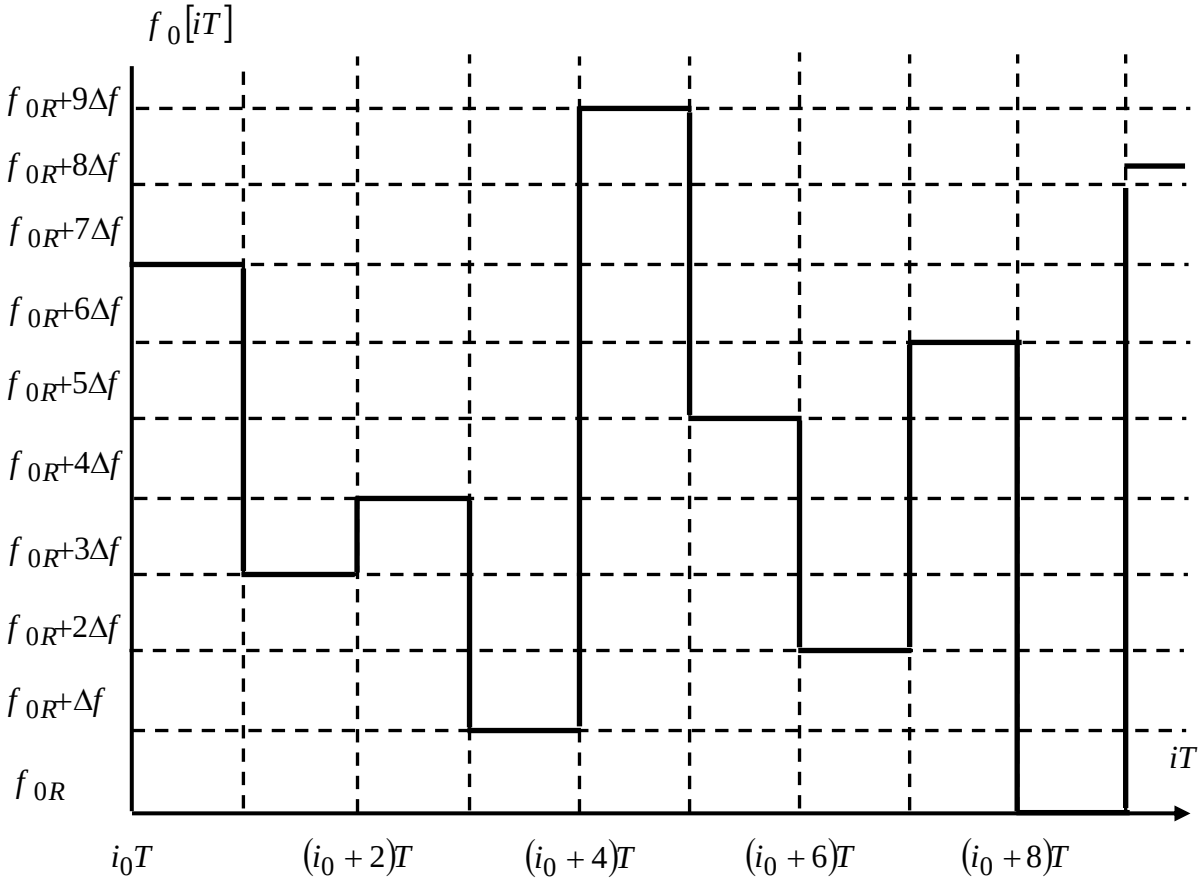


Рис. 2.48. Пример перестройки частоты несущей ФКС в соответствии с выражением (2.40) при выходной последовательности ГПЧ, равной (7, 3, 4, 1, 9, 5, 2, 6, 0, 8)

Естественно, кроме описываемого выражением (2.40), существуют и другие алгоритмы перестройки частоты несущей при FHSS.

Следует отметить, что различают две разновидности перестройки частоты несущей при расширении спектра методом FHSS – *«быструю»* и *«медленную»*, известные также как *быстрое* и *медленное расширение спектра*. Первая из них характеризуется длительностью T такта переключения частоты несущей (см. выражение (2.40)), меньшей, чем длительность битового интервала передаваемой двоичной последовательности. При «медленной» же перестройке частоты значение T в несколько раз больше длительности битового интервала. «Медленная» перестройка более проста в реализации, однако «быстрая» перестройка частоты несущей обеспечивает большую, чем при «медленной», устойчивость к узкополосным помехам, за счет многократной передачи каждого бита на различных частотах.

Основными *преимуществами* метода FHSS по сравнению с другими методами расширения спектра являются простота технической реализации при достаточно высокой (однако несколько меньшей, чем у ряда разновидностей DSSS) степени защиты данных от несанкционированного доступа [3].

К основным *недостаткам* метода FHSS относятся [3, 5]:

- использование в каждый конкретный момент времени только небольшой части диапазона частот, выделенного под ФКС или группу ФКС (в то время как, например, технология DSSS всегда использует весь указанный диапазон);
- меньшая, чем при DSSS, устойчивость к затуханию и эффектам многолучевого распространения сигнала;
- меньшая, чем у DSSS, степень защиты данных от несанкционированного декодирования (иными словами, декодирование данных посторонними лицами при FHSS несколько проще, чем при DSSS).

Следует отметить, что вышеперечисленные преимущества метода FHSS, в первую очередь, важны для систем связи военного назначения. Поэтому исторически первым практическим приложением данного метода являлось его использование войсками связи германской армии во время 1-й Мировой войны. Метод FHSS также широко применялся во время 2-й Мировой войны и в послевоенное время (до начала 80-х гг. прошлого века) в закрытых (секретных) системах связи. В комбинации с другими методами защиты данных FHSS применяется в них и в настоящее время. Основными коммерческими приложениями метода FHSS в настоящее время являются:

- ряд стандартов обмена данными на физическом уровне по ФКС мобильной телефонной связи [5];
- ряд стандартов физического уровня группы IEEE 802.11 (Wi-Fi) [5];
- интерфейс беспроводных микролокальных ВС Bluetooth [3].

В целом, метод FHSS достаточно широко распространен в современных беспроводных ФКС ВС.

2.9.3. Расширение спектра методом DSSS

Данный метод состоит в преднамеренном уменьшении минимальной длительности $T_{\min}$ интервала времени между изменениями информативного параметра сигнала-носителя данных (см. рис. 2.2) при

неизменной длительности битового интервала передаваемой двоичной последовательности [3, 5]. Из выражений (2.3) и (2.4) нетрудно заметить, что уменьшение $T_{\min}$ в N раз приводит к расширению в такое же количество раз спектра сигнала-носителя.

Практически, метод DSSS реализуется заменой (по определенным правилам) каждого из битов подлежащей передаче двоичной последовательности N битами (называемыми также *чипами*), длительность каждого из которых в N раз меньше длительности заменяемого бита. Полученная в результате замены битовая последовательность затем подвергается модуляции каким-либо из описанных в п. 2.6 способов (например, PSK).

Типовым примером расширения спектра методом DSSS является представление каждого из битов передаваемых данных детерминированными или псевдослучайными взаимно ортогональными последовательностями, применяемое при кодовом мультиплексировании (см. подп. 2.8.5). Известны и другие примеры DSSS, например, представление подлежащих передаче нулей и единиц *кодами Баркера* [5], являющимися детерминированными двоичными последовательностями, обладающими, однако, спектром, характерным для шумовых (случайных) сигналов. Примеры кодов Баркера приведены в табл. 2.8.

Таблица 2.8

Примеры кодов Баркера

Разрядность, бит	Кодовая последовательность Баркера, представляющая логическую единицу*
5	+1 +1 +1 −1 +1
7	+1 +1 +1 −1 −1 +1 −1
11	+1 +1 +1 −1 −1 −1 +1 −1 −1 +1 −1
13	+1 +1 +1 +1 +1 −1 −1 +1 +1 −1 +1 −1 +1
*Логический ноль представляется кодовой последовательностью, получаемой путем инверсии кода единицы (т. е. замены «+1» на «-1» и наоборот)	

Благодаря шумоподобным свойствам кодов Баркера, спектр сигнала-носителя данных, подвергнутый расширению таким кодом, отличается высокой равномерностью, что обеспечивает бóльшую устойчивость к узкополосным помехам, чем при неравномерном спектре сигнала-носителя. С другой стороны, детерминированный характер кодов Баркера упрощает декодирование на приемной стороне. Однако они неприменимы для кодового мультиплексирования ФКС.

Естественно, известны и другие способы кодирования битов подлежащего передаче сообщения, применяемые при расширении спектра методом DSSS [5].

Основными *преимуществами* метода DSSS по сравнению с другими методами расширения спектра, являются [3, 5]:

- потенциально наиболее высокая степень защиты данных от перехвата, в том числе широкополосным приемником, обусловленная практической невозможностью их декодирования при неизвестном коде, использовавшемся при представлении данных;
- более высокая, чем при FHSS, устойчивость к затуханию и эффектам многолучевого распространения сигнала;
- возможность разделения несколькими DSSS-ФКС одного и того же участка передающей среды, т. е. их кодового мультиплексирования (см. подп. 2.8.5).

Основным *недостатком* метода DSSS является относительная сложность реализации по сравнению с другими методами расширения спектра. Однако при современном уровне развития аппаратно-программных средств кодирования и передачи данных этот недостаток не является существенным.

В настоящее время метод DSSS широко применяется в системах связи (в основном, беспроводных) как коммерческого, так и военного назначения. Основными коммерческими приложениями метода DSSS являются:

- 3G-системы мобильной связи (в том числе мобильного доступа к Интернет) [7];
- ряд стандартов физического уровня группы IEEE 802.11 (Wi-Fi) [5];
- ряд стандартов физического и канального уровней беспроводных ЛВС группы IEEE 802.15 (ZigBee) [5].

2.9.4. Расширение спектра методом THSS

Данный метод состоит во включении/отключении сигнала-носителя данных ФКС в моменты времени, задаваемые выходным сигналом ГПП (см. рис. 2.32). Расширение спектра сигнала-носителя при этом достигается за счет внесения в него разрывов при коммутации [5].

Метод THSS до настоящего времени не нашел широкого распространения в системах связи, в том числе в ФКС ВС [5].

2.9.5. Расширение спектра методом CSS

Этот метод основывается на представлении данных в ФКС модулированным сигналом, в качестве несущей которого выступает так называемый *chirp-сигнал* (от англ. слова «chirp», дословно переводимого как «чириканье», «щебетанье»). Под таким сигналом в теории связи подразумевается синусоида, частота которой возрастает или убывает во времени по определенному закону. На практике наиболее распространены chirp-сигналы с линейно возрастающей или убывающей частотой (linear chirp), а также с экспоненциальным возрастанием или убыванием частоты (известные под названиями «exponential chirp» или «geometric chirp»). Пример временной диаграммы chirp-сигнала с линейно возрастающей частотой представлен на рис. 2.49.

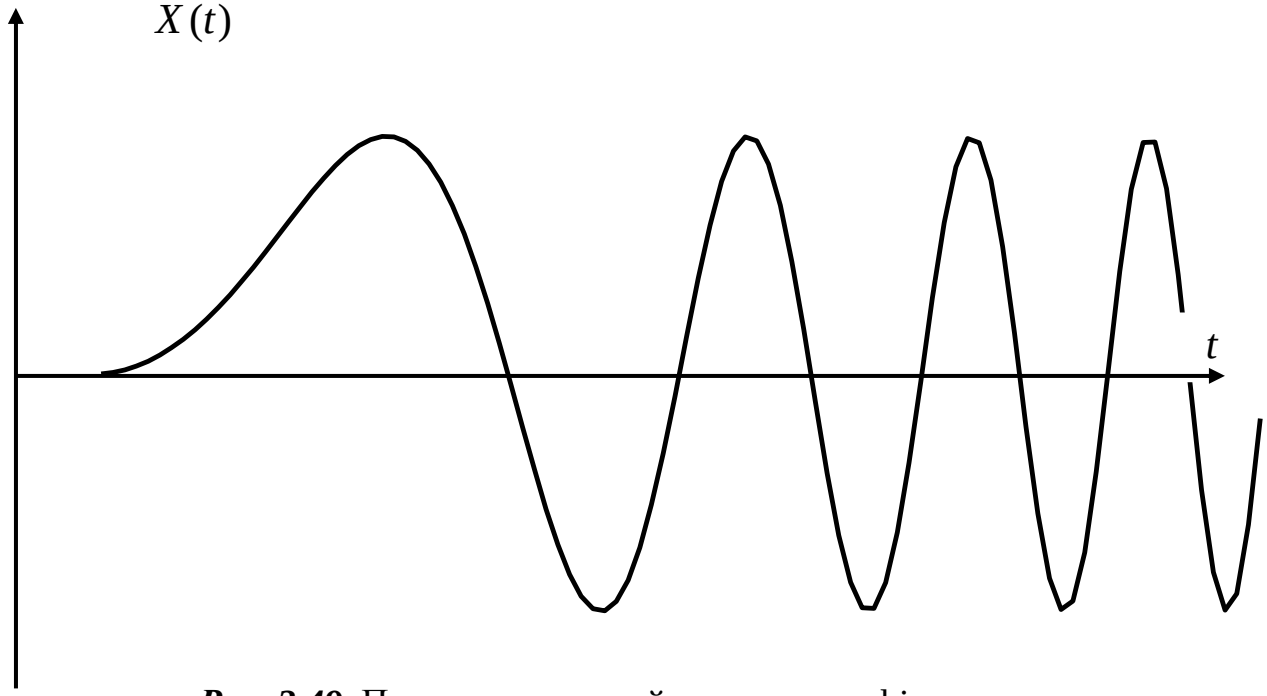


Рис. 2.49. Пример временной диаграммы chirp-сигнала с линейно возрастающей частотой

Основными *преимуществами* метода CSS, по сравнению с другими методами расширения спектра, являются [5]:

- высокая устойчивость к многолучевому распространению сигнала, в том числе при достаточно малой мощности передатчика;
- устойчивость процесса обмена данными к изменениям частоты несущей, вызванным эффектом Доплера (что важно в системах мобильной связи, в том числе мобильного доступа к Интернет);
- простота реализации.

Недостатком метода CSS является слабая защищенность данных от несанкционированного перехвата. Из-за этого недостатка данный метод применяется практически только в ФКС коммерческого назначения. Основным известным приложением метода CSS является ряд стандартов группы IEEE 802.15 (ZigBee) [5].

2.9.6 Техническая реализация расширения спектра

Осуществляется посредством того же сетевого оборудования, что и реализация других функций физического уровня модели OSI (см. подп. 2.5.7, 2.6.6, 2.7.5 и 2.8.6). Принципы построения этого оборудования будут рассмотрены в гл. 3.

Выводы по п. 2.9

Расширение спектра применяется преимущественно в беспроводных ФКС. Его основными целями являются повышение устойчивости к узкополосным помехам и защита передаваемых по ФКС данных от несанкционированного доступа и преднамеренного искажения. Наиболее распространенными на практике методами расширения спектра являются метод прямой последовательности (DSSS), обеспечивающий достижение обеих вышеуказанных целей, а также FHSS и CSS, применяемые в настоящее время в основном для повышения помехоустойчивости.

Выводы по главе 2

Основной функцией физического уровня модели OSI является обеспечение обмена потоками битов между 2-мя узлами ВС по *ФКС*, определяемому как фрагмент коммуникационной системы ВС, обеспечивающий обмен информацией между 2-мя соседними узлами ВС и не содержащий промежуточных устройств коммутации или мультиплексирования. Указанная функция сводится к реализации двух базовых процедур:

- преобразования кадра данных, подлежащего передаче между абонентами некоторого ФКС, в сигнал, пригодный для передачи по соответствующему ФКС;
- обратного преобразования указанного сигнала в исходный кадр данных на приемной стороне.

При этом данные представляются как однородный поток битов, безотносительно к формату и смысловому назначению отдельных групп битов вышеназванного кадра.

ФКС реализуются в соответствии с *обобщенной структурной схемой* (см. рис. 2.1). Основными компонентами ФКС являются: аппаратура передачи данных (АПД, DCE), блоки усиления и регенерации и физическая среда передачи данных (передающая среда). В ее качестве может выступать провод или кабель различных типов, воздух или вакуум. Функции АПД реализуют модемы и/или сетевые адаптеры абонентских компьютеров и аналогичные блоки коммутационного оборудования ВС (коммутаторов, маршрутизаторов и т. п.).

Основными *типами ФКС ВС* являются (см. п. 2.2):

- кабельные абонентские ФКС коммутируемой телефонной сети общего пользования (КТСОП), в настоящее время применяемые преимущественно для доступа физических лиц к ГВС, в первую очередь, к Интернет (в основном – с использованием технологий xDSL);

- магистральные ФКС КТСОП, применяемые также в качестве магистральных линий связи ГВС, сетей мегаполисов и, в ряде случаев, крупных ЛВС (при этом под сетевой трафик, обычно, методами мультиплексирования, выделяется часть пропускной способности МЛС КТСОП);

- кабельные ФКС, специально выделенные для обмена данными между узлами ВС, в первую очередь, – кабельные ФКС ЛВС и магистральные кабельные ФКС ГВС и сетей мегаполисов;

- беспроводные абонентские ФКС системы мобильной телефонной связи и КТСОП, применяемые, как и кабельные АЛС КТСОП, преимущественно для доступа физических лиц к ГВС;

- магистральные, в том числе спутниковые, беспроводные ФКС КТСОП, выступающие, как и кабельные МЛС КТСОП, также в качестве магистральных линий связи ГВС, сетей мегаполисов и крупных ЛВС (путем выделения части пропускной способности под сетевой трафик);

- выделенные беспроводные ФКС, в первую очередь, – беспроводные ФКС ЛВС и магистральные беспроводные ФКС ГВС.

При этом на полосу пропускания кабельных абонентских ФКС КТСОП, а также всех типов беспроводных ФКС накладываются организационно-законодательные ограничения, причем определяемые ими как нижняя, так и верхняя граничные частоты существенно отличают-

ся от нуля. Полоса пропускания других типов ФКС зависит только от технических характеристик их компонентов, узлов и блоков, в первую очередь, – передающей среды.

Наиболее распространенными типами *передающей среды* ФКС являются следующие (см. п. 2.3):

- электрические провода, в настоящее время используемые только в качестве передающей среды АЛС КТСОП (см. рис. 2.6), однако, постепенно вытесняемые и из этой области применения неэкранированными витыми парами (UTP-кабелями), а с недавнего времени – и волоконно-оптическими кабелями (ВОК);

- UTP-кабели, широко применяемые в прокладываемых внутри зданий линиях связи ЛВС протяженностью порядка единиц – десятков метров, а также, с недавнего времени – в АЛС КТСОП;

- экранированные витые пары (STP-кабели), используемые для прокладки кабельных ФКС, требующих повышенной защиты от помех, протяженностью порядка десятков метров (в основном, внутри зданий, реже – вне зданий);

- ВОК, применяемые в основном в прокладываемых вне зданий магистральных ФКС, в том числе в МЛС КТСОП и в выделенных магистральных линиях связи ВС; с недавнего времени применение ВОК становится оправданным и в абонентских ФКС;

- беспроводная передающая среда, применяемая для реализации ФКС при сложности, невозможности или экономической невыгодности прокладки кабельных трасс, а также для связи с ВС мобильных абонентов.

Данные по ФКС передаются в последовательном формате. В качестве *сигналов-носителей данных* выступают электрические напряжения, токи или электромагнитные волны различных частотных диапазонов в зависимости от конкретного типа ФКС. Информацию о передаваемых данных несут изменения во времени состояния сигнала-носителя, которое характеризуется значениями одного или нескольких информативных параметров указанного сигнала (амплитуды, частоты, фазы и т. п.). В ФКС без организационно-законодательных ограничений на полосу пропускания применяются двух- или многоуровневые сигналы-носители, называемые линейными кодами (см. п. 2.5), граничные частоты спектра которых описываются выражениями (2.3). Информацию о передаваемых посредством линейных кодов данных несут уровни или перепады уровней сигнала-носителя. В ФКС с организационно-законодательными ограничениями на полосу пропускания, в том числе на ее нижнюю граничную частоту (к кото-

рым относятся, в первую очередь, беспроводные ФКС и АЛС КТСОП), в качестве носителей данных используются модулированные синусоидальные сигналы, граничные частоты спектра которых определяются выражениями (2.4). При передаче данных по ФКС ВС применяются частотная, дифференциальная фазовая, квадратурная амплитудная и импульсная кодовая модуляция (см. п. 2.6). Из них в настоящее время наиболее распространены различные разновидности дифференциальной фазовой и квадратурной амплитудной модуляции. Современные способы модуляции позволяют представлять одним состоянием модулированного сигнала не один, а несколько битов модулирующей двоичной последовательности. При этом чем больше их количество, тем больше число информативных (различаемых) состояний модулированного сигнала и тем большее отношение «сигнал-шум» в ФКС требуется для корректного детектирования представляемых сигналом двоичных данных (см. табл. 2.6). Поэтому большинство современных модемов являются мультистандартными, с автоматическим выбором способа модуляции и количества различаемых состояний модулированного сигнала в зависимости от текущего отношения «сигнал-шум» в ФКС.

Большинство типов линейных кодов и способов модуляции для корректного детектирования на приемной стороне представляемой ими двоичной последовательности требует ее предварительного *логического кодирования*. Такое кодирование состоит в преобразовании этой последовательности в некоторую другую, непосредственно передаваемую по ФКС методом линейного кодирования или модуляции. На приемной стороне после линейного декодирования или, соответственно, демодуляции, из последовательности, полученной в результате логического кодирования, восстанавливаются исходные двоичные данные. Основными целями применения логического кодирования являются следующие:

- устранение длинных последовательностей нулей и единиц в передаваемой двоичной последовательности, которые могут привести к потере взаимной синхронизации передатчика и приемника;

- обеспечение нулевой или близкой к нулевой постоянной составляющей сигнала-носителя, что необходимо при передаче данных по ФКС на основе электрического кабеля;

- повышение устойчивости процесса обмена данными к ошибкам, обусловленным ограниченной полосой пропускания ФКС и помехами в нем.

Наиболее распространенными методами логического кодирования являются:

- избыточное RLL-кодирование (см. подп. 2.7.2), позволяющее устранять длинные последовательности нулей и единиц, при возможности частичного обнаружения (без исправления) ошибок обмена данными;

- скремблирование (см. подп. 2.7.3), обеспечивающее при отсутствии избыточности эффективное устранение длинных последовательностей нулей и единиц, а также практически нулевое значение постоянной составляющей сигнала-носителя, однако не позволяющее обнаруживать ошибки обмена данными;

- избыточное сверточное кодирование (см. подп. 2.7.4), позволяющее частично обнаруживать и исправлять ошибки обмена данными при отсутствии устранения постоянной составляющей сигнала-носителя и длинных последовательностей нулей и единиц.

При этом, поскольку ни один из этих методов не позволяет полностью решить все вышеперечисленные задачи логического кодирования, на практике обычно применяются их сочетания. Следует отдельно отметить, что сочетание сверточного кодирования на физическом уровне и блочного помехоустойчивого кодирования на канальном уровне (см. рис. 2.34 и пояснения к нему) является эффективным методом обнаружения и устранения ошибок обмена данными по ФКС с относительно высоким уровнем BER (см. подп. 2.1.3), порядка 10^{-3} – 10^{-4}. Таковыми являются, например, беспроводные ФКС, а также АЛС КТСОП при использовании скоростных, но не помехоустойчивых методов модуляции (КАМ и ИКМ). С другой стороны, сверточное кодирование, как правило, не применяется в ФКС с малыми значениями BER, порядка 10^{-5} и менее, например, в выделенных кабельных ФКС.

Во многих практических случаях один и тот же участок передающей среды используется множеством ФКС (в частности, беспроводная передающая среда является разделяемой по самой своей природе). Для устранения взаимного влияния сигналов-носителей данных этих ФКС используются специальные приемы, называемые *мультиплексированием*. В ФКС ВС применяются в основном следующие методы мультиплексирования:

- частотное (FDM) (см. подп. 2.8.2), в настоящее время используемое преимущественно в сочетании с другими методами мультиплексирования; исключение составляет ортогональное частотное мульти-

плексирование (OFDM), применяемое и без сочетания с какими-либо другими методами (в частности, протоколами обмена данными на физическом уровне группы WiMAX);

- волновое (WDM), являющееся, по существу, частным случаем частотного, распространенным в ФКС на основе ВОК (см. подп. 2.8.3);

- временное (TDM) – асинхронное, синхронное и плезиохронное (см. подп. 2.8.4), широко применяемое при обмене данными в цифровой форме как в кабельных, так и в беспроводных ФКС, в том числе в сочетании с другими методами мультиплексирования (FDM, WDM, CDMA);

- кодовое (CDMA) (см. подп. 2.8.5), как правило, применяемое в сочетании с частотным и/или временным (в частности, на таком сочетании методов мультиплексирования основан ряд современных стандартов мобильной связи).

В беспроводных ФКС применяется дополнительный прием повышения эффективности обмена данными – *расширение спектра сигнала-носителя данных* (см. п. 2.9), обеспечивающее:

- повышение устойчивости обмена данными к узкополосным помехам, в том числе генерируемым преднамеренно;

- снижение мощности излучения на единицу частоты (что важно, например, в спутниковых системах связи);

- защиту передаваемых по ФКС данных от несанкционированного перехвата узкополосным приемником;

- возможность использования диапазона частот, выделенного под ФКС с расширенным спектром сигнала-носителя, также несколькими узкополосными ФКС при их минимальном влиянии на широкополосный ФКС (и наоборот).

Наиболее распространенными на практике методами расширения частоты являются: скачкообразная перестройка частоты (FHSS), метод прямой последовательности (DSSS) и линейная частотная модуляция (CSS).

Техническая реализация функций физического уровня модели OSI осуществляется посредством специального сетевого оборудования: модемов и/или сетевых адаптеров абонентских компьютеров, а также функционально аналогичных блоков коммутационных устройств ВС (коммутаторов, маршрутизаторов и т. п.). Поскольку указанное оборудование реализует также функции канального уровня, принципы его реализации будут рассмотрены в гл. 3.

Вопросы для самопроверки

1. Дайте определение ФКС ВС, перечислите его основные компоненты и их функции.

2. Как представляются двоичные данные в ФКС ВС?

3. Поясните, по какой причине в ФКС с организационно-законодательными ограничениями на полосу пропускания в качестве носителей данных используются модулированные синусоидальные сигналы.

4. Дайте определение пропускной способности ФКС. Сопоставьте формулы Шеннона и Найквиста для ее определения. Обоснуйте, по какой причине вторая из них имеет больший практический смысл.

5. Проанализируйте, какой из рассмотренных в пп. 2.5 и 2.6 способов линейного кодирования и модуляции потенциально может обеспечить максимально возможную пропускную способность, определяемую формулой Найквиста.

6. Каков смысл параметра BER ФКС? Какие типы ФКС характеризуются минимальным BER? Максимальным BER? Ответ обоснуйте.

7. Чему равно отношение мощности синусоидального сигнала с некоторой частотой на выходе и на входе отрезка передающей среды некоторой длины, если затухание данного типа передающей среды при соответствующих частоте сигнала и длине отрезка равны: 15, 36, 48 дБ?

8. Чему равно затухание отрезка передающей среды с некоторой длиной на некоторой частоте, если отношение мощности синусоидального сигнала с соответствующей частотой на выходе и на входе этого отрезка равны: 0,7; 0,3; 0,08?

9. Поясните смысл параметров NEXT и FEXT передающей среды. По какой причине эти параметры не нормируются для волоконно-оптического кабеля?

10. Поясните физический смысл согласования электрического кабеля.

11. Перечислите функции основных блоков КТСОП (см. рис. 2.6).

12. На чем основывается ограничение полосы пропускания АЛС КТСОП значениями 300 и 3400 Гц?

13. Поясните смысл технологии ADSL (см. рис. 2.7 и 2.8).

14. Обоснуйте применение микроволнового частотного диапазона в беспроводных ФКС ВС. По какой причине выделение под ФКС

диапазона частот шириной 200 кГц при частоте несущей, равной 1 ГГц, вполне возможно, а ее частоте, равной 1 МГц – практически невозможно?

15. Перечислите функции основных блоков СМТС (см. рис. 2.9).

16. Обоснуйте существенное возрастание возможной дальности между соседними ретрансляционными радиостанциями при использовании искусственного спутника связи в качестве промежуточного ретранслятора.

17. В чем заключается различие между электрическим проводом и электрическим кабелем?

18. Поясните представленную на рис. 2.13, *б* схему подключения витой пары при дифференциальной передаче сигнала.

19. Обоснуйте, по какой причине кабель, соединяющий двух абонентов ФКС, должен зануляться только на одной стороне.

20. Поясните принцип полного внутреннего отражения в оптическом волокне.

21. Чем обусловлена меньшая достижимая скорость передачи данных по многомодовому оптическому волокну по сравнению с передачей по одномодовому?

22. Поясните эффект многолучевого распространения сигнала в беспроводных ФКС.

23. Чем обусловлено более высокое значение BER у беспроводных ФКС по сравнению с кабельными?

24. Поясните смысл условия (2.14).

25. Определите максимальную скорость передачи данных (в бит/с) по ФКС с шириной полосы пропускания, равной 10 МГц, при их представлении: кодом NRZI, манчестерским кодом, линейным кодом 2B1Q.

26. Обоснуйте, по какой причине представление данных манчестерским кодом не требует предварительного логического кодирования.

27. По какой причине линейное кодирование не применяется в беспроводных ФКС и в АЛС КТСОП?

28. Оцените примерное значение максимальной скорости передачи данных (в бит/с) по ФКС с шириной полосы пропускания, равной 200 кГц, при их представлении посредством DBPSK, DQPSK, QAM-16, QAM-256, PAM-64, PAM-256 (в предположении, что специальные меры по повышению скорости передачи не применяются).

29. Оцените примерные значения граничных частот DQPSK-сигнала, QAM-16-сигнала и QAM-256-сигнала при частоте несущей 100 МГц и скорости передачи данных, равной 1 Мбит/с (в предположении, что специальные меры по повышению скорости передачи не применяются).

30. Какова скорость передачи информативных данных по ФКС с шириной полосы пропускания, равной 100 МГц, при их представлении кодом NRZI с предварительным 4B/5B-кодированием? С предварительным скремблированием?

31. Поясните смысл совместного применения сверточного и блочного помехоустойчивого кодирования (см. рис. 2.34).

32. По какой причине сверточное кодирование практически не применяется в кабельных ФКС ЛВС?

33. Определите ширину частотного диапазона, который может быть выделен под один частотный подканал OFDM-ФКС, если ширина диапазона частот, выделяемого под ФКС в целом, равна 5 МГц, а число поднесущих равно 512.

34. Какова суммарная скорость передачи данных по OFDM-ФКС с характеристиками, приведенными в подп. 2.8.2, при использовании DQPSK для представлении данных в каждом из частотных подканалов?

35. Определите максимальную скорость передачи данных по CDMA-ФКС с шириной полосы пропускания, равной 5 МГц, разрядностью идентифицирующей последовательности, равной 64-м битам, и использованием DBPSK для представления данных. Предполагается, что специальные меры для повышения скорости обмена не применяются.

36. Во сколько раз расширяется спектр сигнала-носителя данных при применении 7-битового кода Баркера? CDMA с 64-битовой идентифицирующей последовательностью?

3. КАНАЛЬНЫЙ УРОВЕНЬ

3.1. Назначение канального уровня

Канальный уровень (Data Link Layer – DLL) пересылает данные от узла к узлу в пределах одного физического канала связи [2, 3, 7]. Протоколы канального уровня получают пакеты данных от сетевого уровня, инкапсулируют их в кадры и передают кадры на физический уровень. Канальный уровень в модели OSI стоит на втором месте над физическим уровнем, соответственно, канальный уровень называют также Layer 2, или сокращенно L2.

Обратимся к рис. 3.1. Представим себе, что все узлы, показанные на рисунке, могут обмениваться данными. За доставку данных между различными узлами отвечают службы нескольких уровней модели OSI. В частности, канальный уровень передаёт кадры внутри физических сегментов, обозначенных на рисунке как «Сегмент 1», «Сегмент 2», «Сегмент 3», «Сегмент 4». Указанные сегменты могут быть совершенно разнородными. Например, «Сегмент 1» может быль образован витыми парами, узлы в «Сегменте 2» обмениваются данными при помощи радиоволн, «Сегмент 3» и «Сегмент 4» – оптоволоконные. В каждом из сегментов могут использоваться свои собственные протоколы канального уровня.

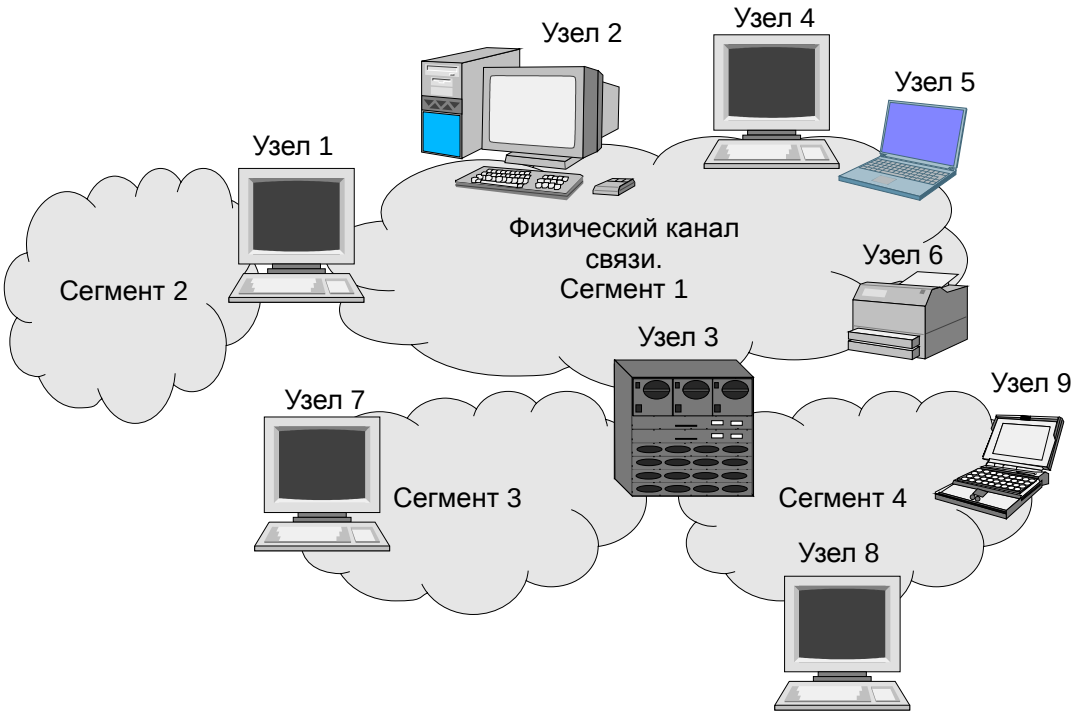


Рис. 3.1. Пример локальной вычислительной сети

Важно отметить, что с точки зрения служб канального уровня, все узлы являются равноправными, не важно, какую именно роль выполняет каждый из узлов. Узел 3 может быть, например, маршрутизатором, Узел 6 – принтером; на первом узле установлена ОС Free BSD, а на пятом – Windows. Канальный уровень ничего об этом «не знает».

Узлы подключаются к каналу передачи данных при помощи сетевых адаптеров (рис. 3.2). Каждый адаптер имеет уникальный сетевой адрес.

Рис. 3.2. Подключение сетевого адаптера

Канальный уровень формирует кадр данных, состоящий в общем случае из физического адреса получателя, физического адреса отправителя, пакета данных, контрольной суммы (рис. 3.3) [3, 7].

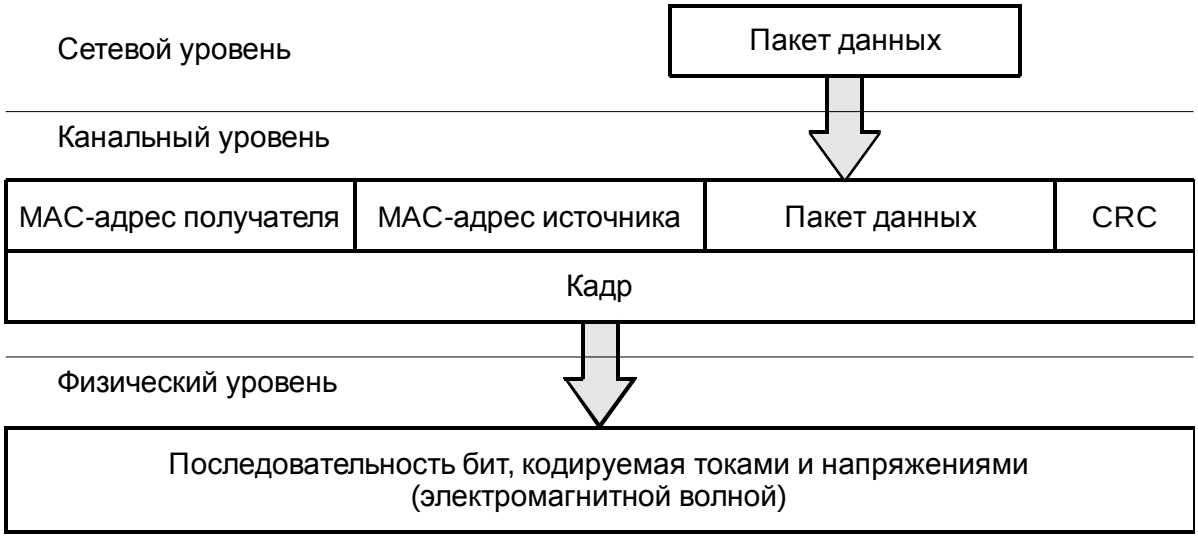


Рис. 3.3. Инкапсуляция данных в кадр

Некоторые протоколы канального уровня отвечают за безошибочную пересылку пакета данных в пределах физического сегмента сети,

поэтому в ряде случаев предусматривается повторная посылка ошибочного кадра. Следует отметить, что целостность данных канальным уровнем поддерживается только локально. Например, информация в Интернет при движении от сервера к пользователю может пройти множество разнородных сетей. Целостность данных при этом гарантирует транспортный уровень.

Протоколы канального уровня строятся исходя из особенностей физической передачи данных и тесно связаны с протоколами физического уровня. Услуги канального уровня в большинстве случаев реализуются аппаратно непосредственно сетевым адаптером. Адаптер является полуавтономным устройством. Он самостоятельно выполняет действия по организации передачи пакетов сетевого уровня от узла к узлу. В качестве среды передачи широко используются витая пара, оптоволокно, радиоэфир (см. п. 2.3).

Часто несколько узлов используют единую физическую среду для передачи данных. При этом каждый узел «слышит» посылки любого другого узла. Понятно, что если посылать данные будут несколько узлов одновременно, то все попытки что-то передать будут обречены на провал. Таким образом канальный уровень должен также обеспечить доступ к физической среде передачи данных. Соответствующие протоколы выделяют в отдельный подуровень управления доступом к среде (Media Access Control – MAC).

3.2. Задачи канального уровня

Протоколы канального уровня решают следующие задачи [1, 3, 7]:

- Формирование кадров. Структура кадра зависит от конкретного используемого протокола.
- Управление доступом к физической среде передачи данных. Сложность данной задачи варьируется от простейшего случая дуплексной связи двух узлов до обеспечения взаимодействия узлов в радиоэфире.
- Обнаружение ошибок при передаче данных. Иногда протоколы канального уровня обеспечивают также исправление ошибок. Следует отметить, что надёжная доставка данных канальным уровнем работает локально, то есть только в пределах одной физической сети.
- Управление потоками данных. Например, с целью предотвращения переполнения входного буфера сетевого адаптера.

3.3. Примеры технологий и стандартов канального уровня

Типовыми примерами стандартов канального уровня являются следующие [3, 7]:

Ethernet (группа стандартов IEEE 802.3)

Технология Ethernet (буквальный перевод *«сеть в эфире»*) на сегодняшний день наиболее широко используется при организации локальных сетей ЭВМ. В качестве физической среды передачи данных может быть использован коаксиальный кабель, витая пара, оптическое волокно.

Wi-Fi (Группа стандартов IEEE 802.11)

Набор стандартов IEEE 802.11 включают в себя протоколы, ориентированные на построение беспроводных сетей. Адаптеры такой сети оборудованы радиопередатчиком и радиоприёмником. Сетевое оборудование, построенное по стандартам IEEE 802.11, распространяется под торговой маркой «Wi-Fi».

PPP

Протокол PPP (Point to Point Protocol) – протокол соединения двух узлов. Содержит целое семейство протоколов, применяемых при соединении двух узлов. В числе прочих включает протоколы проверки пароля и протоколы управления линией связи.

ARP

Протокол ARP (Address Resolution Protocol) – протокол разрешения адресов. Предназначен для получения сетевыми адаптерами таблиц соответствия сетевых адресов (MAC) IP-адресам. ARP позволяет узнать, какому узлу принадлежит тот или иной IP-адрес. Существует также реверсивный протокол разрешения адресов, позволяющий по известному MAC получить IP-адрес.

CSMA, CSMA/CD, CSMA/CA

Технология CSMA (Carrier Sense Multiple Access) – множественный доступ с контролем несущей, CSMA/CD (Carrier Sense Multiple Access / Collision Detection) – множественный доступ с контролем несущей с обнаружением коллизий, CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance) – множественный доступ с контролем несущей с предотвращением коллизий кадров. Данные технологии применяются для организации совместной работы с временным разделением множества узлов, подключенных к одной физической среде.

3.4. Технологии CSMA, CSMA/CD, CSMA/CA

Каким образом можно разделить пропускную способность единой физической среды между отдельными узлами? Отметим, что хотелось бы добиться следующих характеристик передачи данных. Пусть R (бит в секунду) – пропускная способность канала связи, образованного с использованием данной среды. Пусть N – количество узлов, одновременно использующих физическую среду. При работе на передачу одного узла, он должен целиком использовать пропускную способность R. При одновременной работе на передачу $M \leq N$ узлов, на каждый узел должна приходиться пропускная способность канала R/M.

Вспомним существующие в многоканальных системах связи следующие способы разделения каналов:

1) частотное;
2) временное;
3) по форме сигнала (кодовое разделение);
4) по характеристикам электромагнитной волны (включая разделение по поляризации и по направлению распространения);
5) комбинированное, например, частотно-временное разделение;

Частотное разделение каналов вполне оправдано и принципиально необходимо в радиоэфире. Здесь нужно позаботиться об электромагнитной совместимости сети связи. Передатчики должны излучать радиосигналы только в пределах ограниченной полосы частот, а приёмники – принимать радиосигналы только в этой полосе. Использование частотного разделения в физических средах, образованных такими направляющими системами, как двухпроводная линия связи или витая пара, целесообразно в тех случаях, когда на этих средах функционирует разнородное оборудование (например, аналоговая телефонная связь плюс DSL-модем).

В случае подключения к витой паре нескольких равноправных узлов (с точки зрения канального уровня модели OSI), наиболее приемлемо временное разделение, которое может быть осуществлено с помощью трёх технологий:

1. Технология Demand Priority Access – доступ по приоритету запроса. Используется топология «звезда». Применение кабелей с несколькими витыми парами позволяет реализовать полнодуплексный режим связи. Управление доступом к физической среде осуществляется концентратором, который опрашивает узлы и получает запросы

на передачу данных. В запросе указывается приоритет данных. Концентратор на основе анализа приоритетов выполняет соединение между узлом-источником и узлом получателем.

2. Технология Token Ring. Логически узлы объединены в кольцо. Соседний узел может передавать следующему в кольце специальный кадр, именуемый *маркером*. Узел, «владеющий» в данный момент маркером, монопольно использует среду передачи. Если узел имеет данные, подготовленные к отправке, он передаёт один или несколько кадров данных, потом пересылает маркер следующему узлу в кольце. Если узлу нечего передавать, то он просто пересылает маркер дальше по кольцу.

Выгоды технологии Token Ring очевидны. Для сети, находящейся в исправном состоянии, имеется гарантированное максимальное время задержки передачи данных. Сложности возникают при отказе одного из узлов кольца. Также необходимы специальные протоколы включения узла в кольцо и отключения узла из кольца. Нужно обратить внимание на то, что если кольцо организовано физически (а не логически), надо принимать специальные меры для поддержания функционирования сети при выходе узла из строя. Накладные расходы на пересылку маркера также снижают эффективность технологии Token ring.

3. Технология CSMA (Carrier Sense Multiple Access) – множественный доступ с контролем «несущей». Рассмотрим эту технологию подробнее. На сегодняшний день широко используются две модификации – доступ с обнаружением коллизий (CSMA/CD) и доступ с предотвращением коллизий (CSMA/CA).

Идеи, положенные в основу технологии CSMA, просты. Во-первых, прежде чем начать передавать данные, узел прослушивает физическую среду. Передача начинается, только тогда, когда все остальные узлы «молчат». Во-вторых, узел не должен «монополизировать» физическую среду. «Сказал» сам – дай «сказать» другому. В-третьих, во время передачи узел также прослушивает среду, чтобы обнаружить *коллизию* – наложение физических сигналов от нескольких узлов. При наличии коллизии передаваемые данные искажаются и передачу необходимо повторить.

Почему вообще возникают коллизии, если узел прослушивает среду, прежде чем начать передачу? На рис. 3.4 показано, как распространяется сигнал в физической среде передачи, если передаёт только один узел (в данном случае – первый).

Видно, что распространение сигнала занимает конечное время. На рис. 3.4 также показан межкадровый интервал IFG – Inter Frame Gap, в течение которого другие узлы могут начать передачу.

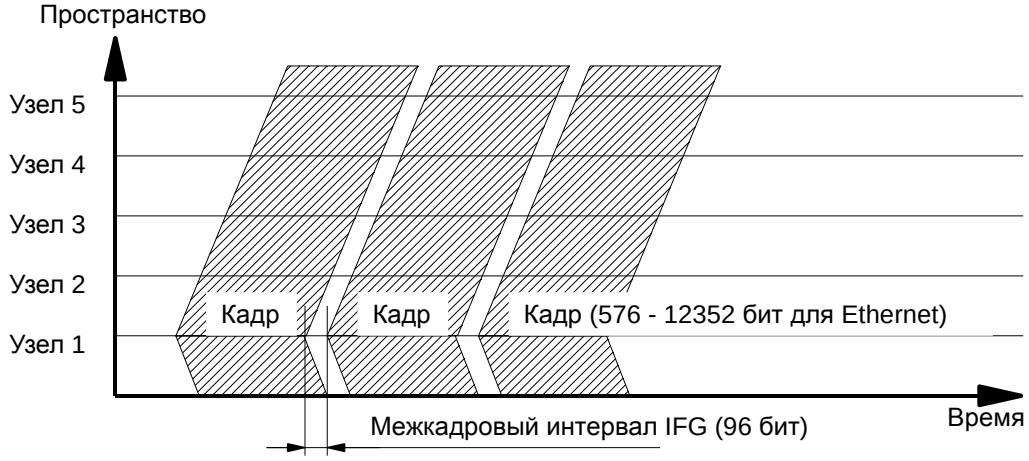


Рис. 3.4. Распространение данных в физической среде передачи [8]

Далее обратимся к рис. 3.5. В момент времени t_1 передачу начал первый узел, в момент времени t_2 четвёртый узел прослушал среду передачи и также начал передавать. В момент t_3 коллизия возникла возле четвёртого узла, в момент t_4 коллизия возникла возле первого узла. Кадры, передаваемые первым и четвёртым узлом, будут потеряны.

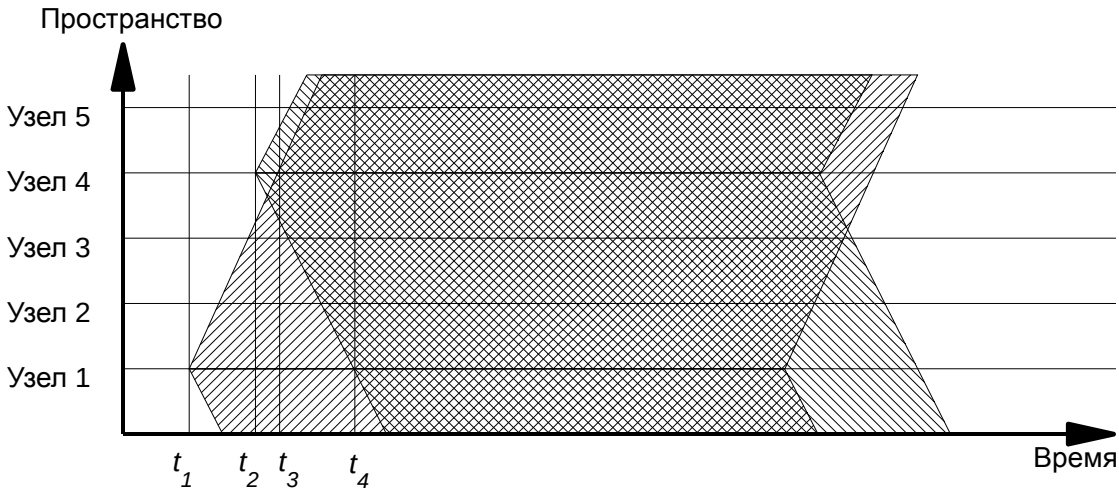


Рис. 3.5. Возникновение коллизии [8]

Для экономии времени и повышения эффективности использования разделяемой физической среды необходимо прервать передачу кадра почти сразу после обнаружения коллизии (рис. 3.6). Несколько

бит передаётся после обнаружения коллизии для того, чтобы остальные узлы также обнаружили, что среда занята и не пытались начать передачу.

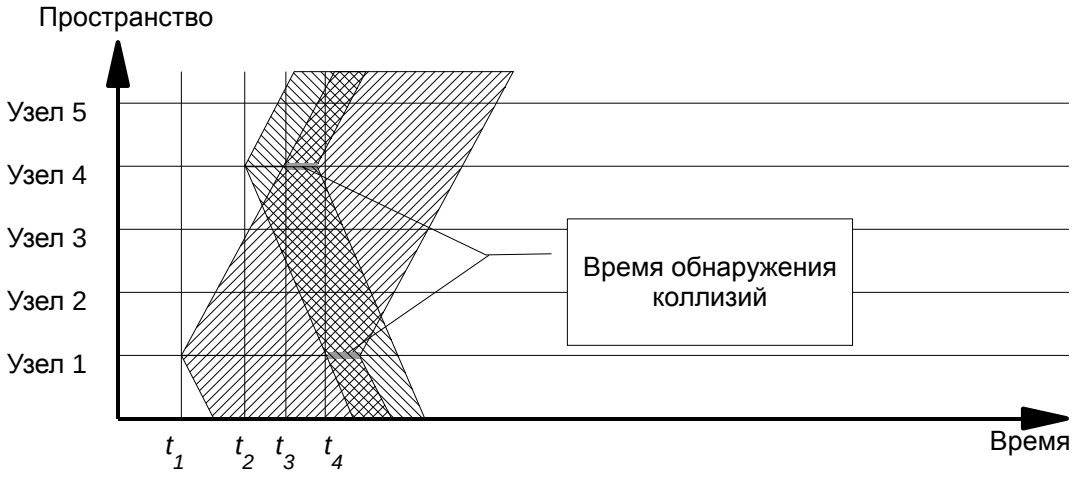


Рис. 3.6. Возникновение коллизии и прерывание передачи [8]

Что делать после коллизии узлу, который имеет непереданный кадр? Нужно немного подождать и опять попробовать начать передачу. Сколько ждать? После обнаружения коллизии узел переходит в фазу *экспоненциального отката*. Если коллизия для данного кадра была обнаружена впервые, то выбирается случайным образом число из ряда (0, 1), это число умножается на время, за которое может быть передано 512 бит (Time Slot – интервал времени). Узел сразу повторяет попытку передачи или ожидает один интервал.

После второй неудачи случайное число выбирается из ряда (0, 1, 2, 3). Узел может выжидать до трёх интервалов. После третьей попытки имеется ряд (0, 1, 2, 3, 4, 5, 6, 7). И так далее. В десятой и последующих попытках случайное число выбирается из ряда (0 .. 1023). После 16-ти попыток сетевой адаптер извещает вышестоящий уровень OSI об ошибке сети. На рис. 3.7 представлен примерный алгоритм работы сетевого адаптера по технологии CSMA/CD.

Беспроводные сети (Wireless Networks – WLAN) имеют свои особенности. В нормально функционирующей сети, построенной с использованием направляющих систем, все узлы, входящие в физический сегмент, принимают сигналы друг друга. В радиосети это правило не выполняется, существует *проблема скрытого узла*. Рис. 3.8 иллюстрирует ситуации, при которых узел *2* способен обмениваться

данными с узлами *1* и *3*, узлы же *1* и *3* не «слышат» друг друга. В первом случае это обусловлено наличием препятствия для распространения радиоволн.

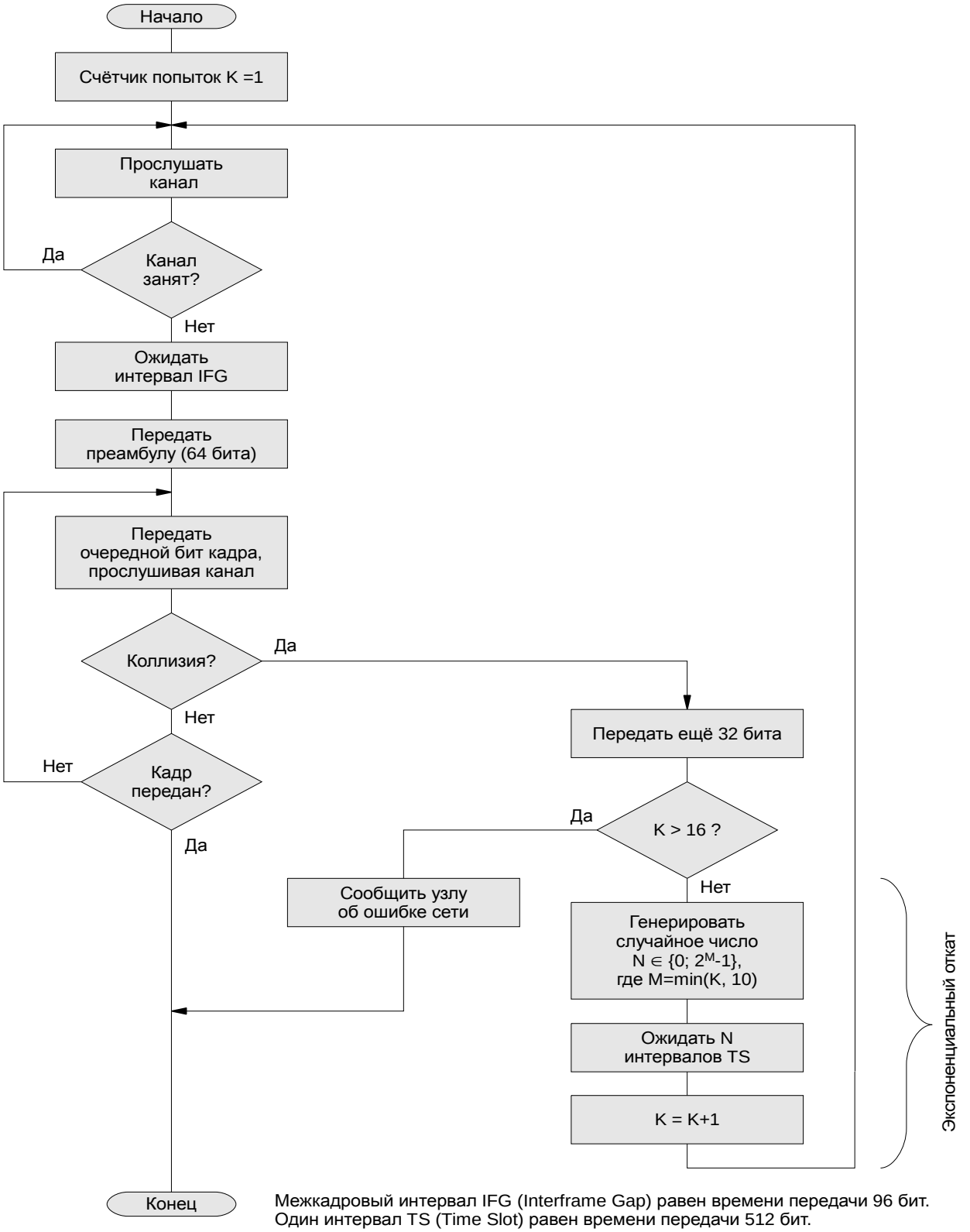
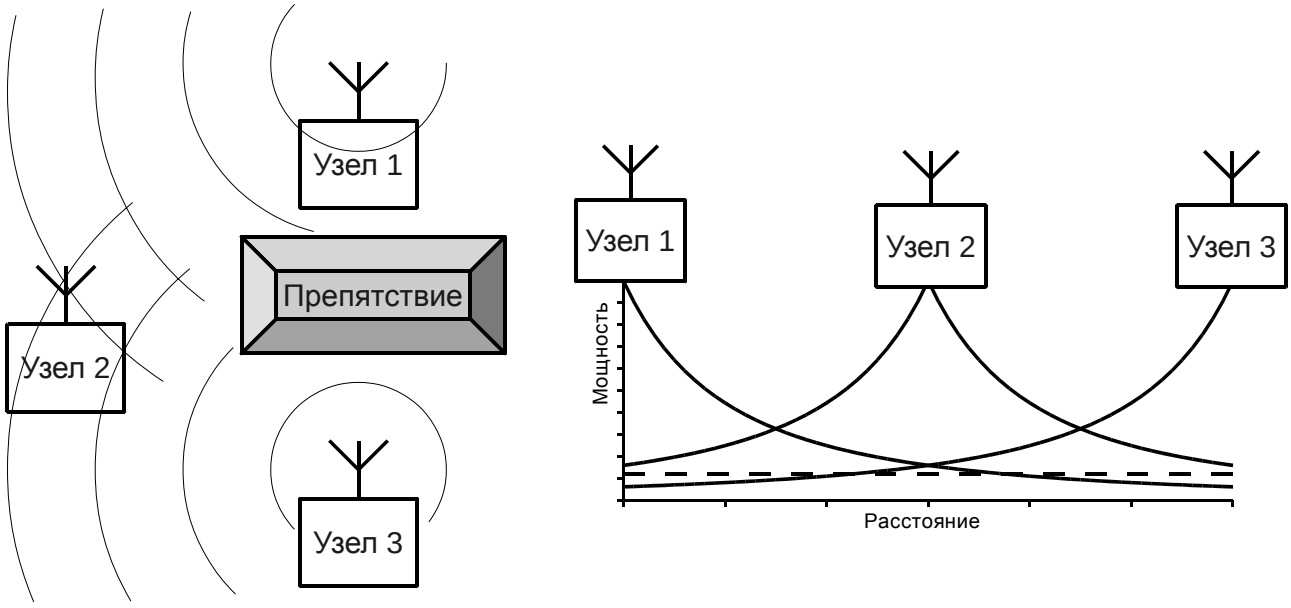


Рис. 3.7. Алгоритм работы сетевого адаптера по технологии CSMA/CD [2, 3]

Во втором – таким расстоянием между узлами *1* и *3*, что мощность сигнала оказывается ниже порогового уровня, необходимого для работы приёмника.



а *б*

Рис. 3.8. Проблема скрытого узла [2, 3]:
а – при наличии препятствия для радиоволн;
б – с учетом падения мощности при удалении от передатчика

В сетях с радиодоступом применяется технология CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance) – множественный доступ с контролем несущей с предотвращением коллизий кадров. Разберём диаграмму, изображённую на рис. 3.9.

Отметим, что в соответствии со стандартами беспроводных сетей, кадры содержат поле длительности. Таким образом узлы, принимающие передачу, могут определить время занятости канала.

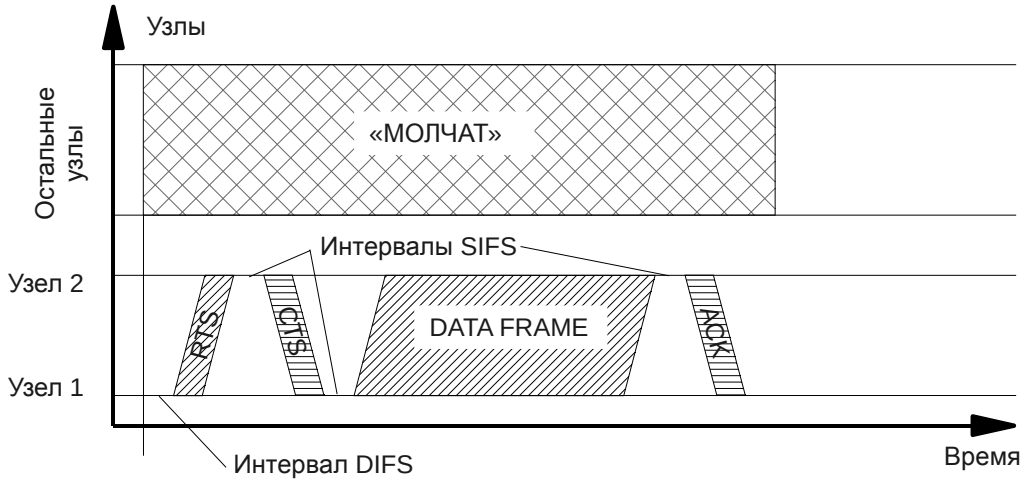


Рис. 3.9. Особенности передачи кадров в радиоэфире [2, 3]

Узел *1* «хочет» передать данные узлу *2*. Узел *1* прослушивает эфир в течение интервала DIFS (Distributed coordination function Interframe Space) – межкадрового интервала для протокола распределённой функции координации. При наличии сигнала от других узлов, узел *1* откладывает передачу. Время паузы определяется как произведение случайного числа на длительность интервала Time Slot – стандартного интервала времени (табл. 3.1).

Таблица 3.1

Длительности межкадровых интервалов

Стандарт	Time Slot, мкс	DIFS, мкс	SIFS, мкс
IEEE 802.11b	20	50	10
IEEE 802.11a	9	34	16
IEEE 802.11g	9 или 20	28 или 50	10

Если в течение интервала DIFS узел *1* не обнаруживает мешающих сигналов, то он начинает передачу. Посылка каждого кадра выполняется через интервал времени SIFS (Short Interframe Space) – короткий межкадровый интервал.

Вначале узел *1* передаёт короткий кадр RTS (Request To Send) – запрос на отправку. Узел *2* отвечает коротким кадром CTS (Clear To Send) – разрешение отправки. Теперь узел *1* передаёт кадр данных. Если узел *2* принял данные и проверил CRC, то он отправляет кадр ACK (Acknowledgement Frame) – подтверждение. Явное подтверждение получения данных требуется, так как в сетях с радиодоступом нет гарантии, что узел *2* на протяжении всего кадра данных «слышит» узел *1*. Кадры RTS/CTS и ACK не предаются в случае широковещательной посылки. Кадры RTS/CTS также не передаются, если размер кадра данных мал.

Передача кадров RTS/CTS в большинстве случаев позволяет предотвратить коллизии. Длительность кадров RTS/CTS мала, и вероятность коллизии во время их передачи минимальна. Узлы, находящиеся в окружении узла-получателя, принимают кадр CTS и ожидают указанный в поле длительности промежуток времени. Так решается проблема скрытого узла.

3.5. Технология Ethernet

Ethernet на сегодняшний день является самой популярной технологией построения локальных компьютерных сетей [2, 3, 7]. Группа

стандартов Ethernet описывает физические характеристики среды передачи данных, виды и параметры сигналов, характеристики физических интерфейсов. На канальном уровне определяются форматы кадров и протоколы доступа к физической среде передачи данных. Ethernet может использовать разные физические среды для распространения сигнала – коаксиальный кабель, витую пару, оптоволокно.

Ethernet базируется на технологии CSMA. Перед передачей кадра узел обязательно прослушивает «эфир», и только если канал свободен, – начинает передавать. Существует множество вариаций, обеспечивающих передачу данных со скоростями от 10 Мбит/с до 100 Гбит/с. Во всех случаях на канальном уровне используется *кадр Ethernet.*

Структура кадра Ethernet представлена на рис. 3.10. Заметим, что существует несколько разновидностей кадра, продвигаемых различными фирмами. Остановимся на кадре Ethernet II (DIX).

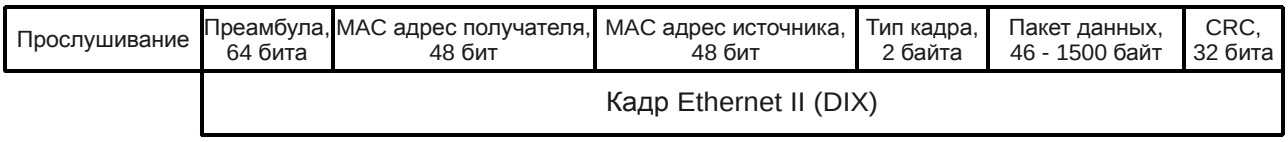


Рис. 3.10. Структура кадра Ethernet

- Перед информационными полями кадра передаётся *преамбула*, представляющая собой фиксированную последовательность нулей и единиц:

10101010101010101010101010101010101010101010101010101010101011.

Две последние единички указывают на то, что далее следует адрес получателя. Необходимость преамбулы связана с работой канала связи на физическом уровне. Во время приёма преамбулы сетевые адаптеры выполняют синхронизацию с частотой и фазой посылаемых битов. Пока происходит синхронизация, часть преамбулы может быть потеряна, но это никак не повлияет на правильный приём следующих бит в кадре.

- Поле «*адрес получателя*» содержит 48-разрядный MAC-адрес получателя. Кадр обрабатывает только тот сетевой адаптер, чей MAC-адрес указан в этом поле. Поле может содержать широковещательный адрес, тогда кадр обрабатывается всеми адаптерами. Также кадр с широковещательным адресом передаётся на все порты коммутаторов.
- Поле «*адрес источника*» содержит 48-разрядный MAC-адрес источника.
- Поле «*тип кадра*» указывает на протокол, инкапсулированный в кадре.

- Поле *данных* имеет переменную длину от 46 до 1500 байт. Значение 1500 байт – это максимальная единица передачи в сети Ethernet (MTU – Maximal Transfer Unit). В ряде случаев пакет данных имеет размер больший, чем 1500 байт. Тогда пакет разбивается вышестоящим протоколом на несколько частей. Если размер полезных данных меньше, чем 46 байт (это часто бывает в протоколе ARP), то оставшиеся байты заполняются произвольными данными.
- Поле *CRC* служит для проверки правильности приёма кадра.

Технология Ethernet продолжает бурно развиваться. В июне 2010 года утверждён стандарт IEEE 802.3ba, описывающий Ethernet со скоростями передачи до 100 ГБит/с (по четырём линиям со скоростями 25 ГБит/с). Максимальное расстояние между узлами при использовании одномодового оптоволокна составляет 10 км.

3.6. Сетевые адаптеры и физические адреса

Все узлы, вне зависимости от их типа и выполняемых ими функций, взаимодействуют с физической средой передачи данных посредством сетевых адаптеров. Адаптер физически является набором электронных компонентов (микросхем), связанным с системными шинами узла с одной стороны, и физической средой передачи данных – с другой.

Сетевой адаптер принимает и передаёт физические сигналы, распространяющиеся в среде передачи данных. Это могут быть немодулированные посылки в двухпроводной линии, модулированные сигналы в двухпроводной линии, радиосигналы, световые импульсы в оптоволокне.

Адаптер является полуавтономным устройством. Получив пакет данных от узла (с точки зрения модели OSI – от вышестоящего протокола в стеке), адаптер самостоятельно выполняет инкапсуляцию пакета в кадр и передачу кадра в канал связи. Получив кадр из канала связи, адаптер проверяет (если это определено протоколом) целостность кадра, извлекает пакет данных и передаёт этот пакет узлу, формируя аппаратное прерывание [2, 7].

Важно отметить, что большинство протоколов канального уровня не подразумевают надёжную передачу данных. Сетевой адаптер, получив «испорченный» кадр с неверной контрольной суммой, просто

отбрасывает его, не извещая о потери кадра вышестоящий уровень. Таким образом вышестоящие протоколы обязательно должны контролировать целостность данных.

Сетевой адаптер «ничего не знает» о правильной последовательности пакетов. Адаптер принимает кадры и передаёт содержащиеся в них пакеты в той последовательности, в которой кадры приходят на вход адаптера. Пакеты данных могут идти по разным маршрутам и перемешиваться. Протоколы вышестоящего уровня должны контролировать правильную последовательность пакетов.

В разделяемой среде передачи данных адаптеры принимают все кадры, распространяющиеся в этой среде. При этом каждый адаптер имеет уникальный физический адрес и обрабатывает кадры, предназначенные только для него. Физический адрес также называют MAC-адресом. Конечно, пользователь может в большинстве случаев изменить физический адрес адаптера, и теоретически возможно подключение к одной физической среде нескольких адаптеров с одинаковыми физическими адресами, однако это нарушит нормальную работу большинства протоколов канального уровня.

Структура MAC-адреса показана на рис. 3.11. MAC-адрес содержит 48 бит. Записывать MAC-адреса принято в шестнадцатеричной нотации.

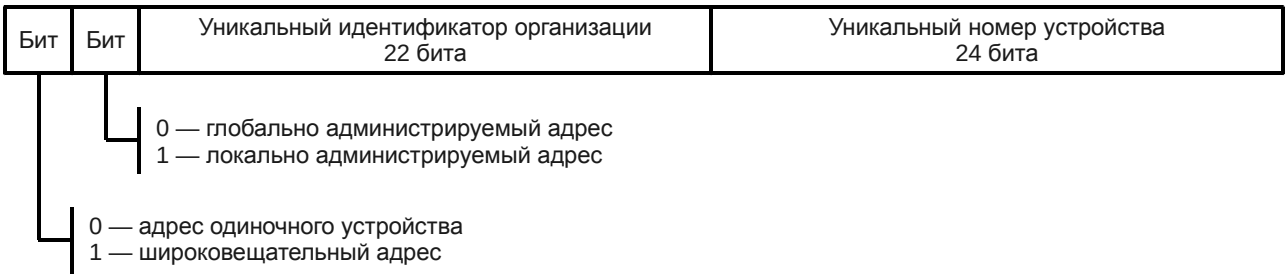


Рис. 3.11. Структура MAC-адреса [3]

Если старший бит адреса установлен в единицу, то это *широковещательный адрес*. Кадры с таким адресом должны принимать и обрабатывать все сетевые адаптеры. В качестве широковещательного чаще всего используется адрес FF:FF:FF:FF:FF:FF. Если старший бит адреса установлен в ноль, то это адрес одиночного устройства. Кадры с таким адресом обрабатываются только соответствующим адаптером.

Если второй по старшинству бит адреса установлен в единицу, то это *локально администрируемый адрес*, то есть адрес, установленный локальным программным обеспечением (операционной системой) или пользователем (посредством соответствующих утилит). Если второй

по старшинству бит адреса установлен в ноль, то это *глобально администрируемый адрес*, то есть уникальный адрес, установленный производителем адаптера.

Для MAC-адреса, установленного производителем, старшие 24 бита (три байта) являются *уникальным идентификатором организации* OUI – Organizationally Unique Identifier. Каждая фирма-производитель получает от института IEEE свои уникальные идентификаторы.

Рассмотрим, например, адрес 00:18:AF:82:83:07. Обратившись к одному из сайтов http://www.networkcenter.info/inform/mac, http://www.coffer.com/mac_find или к официальной страничке IEEE http://standards.ieee.org/develop/regauth/oui/public.html, можно выяснить, что в данном случае производителем адаптера (или микросхемы, на которой базируется адаптер) является компания «Samsung Electronics Co., Ltd».

Оставшиеся 24 бита в случае MAC-адреса производителя являются уникальным номером устройства.

Итак, изначально каждый сетевой адаптер имеет уникальный адрес. Может возникнуть вопрос, зачем нужна многоуровневая система адресации – IP-адреса, MAC-адреса? Не достаточно ли только «плоской» системы физических адресов?

Во-первых, над канальным уровнем может работать не только IP, но и другие протоколы. Во-вторых, представьте себе, что требуется замена адаптера, при этом информация о новом физическом адресе должна быть доступна всем устройствам, выполняющим маршрутизацию в глобальной сети. Это потребует больших накладных расходов. Ещё хуже представляется ситуация с широковещательными рассылками. А что будет, если вышедшей из строя адаптер начнёт непрерывно посылать широковещательные пакеты?

Казалось бы, что можно обойтись совсем без MAC-адресов, ведь тогда каждый адаптер в физическом сегменте сети будет обрабатывать каждый кадр и передавать его на верхний уровень. Но это значительно снизит производительность вычислительных систем. Кроме того, использование в кадрах физических адресов позволяет использовать такие устройства, как коммутаторы.

3.7. Концентраторы, коммутаторы, маршрутизаторы

Физические линии связи между компьютерами могут быть построены с использованием таких технологий, как коаксиальный ка-

бель, витая пара, оптоволокно, радиодоступ. Рассмотрим наиболее широко распространённую в настоящее время технологию витой пары. Для соединения трёх и более узлов в сеть требуется промежуточное устройство (рис. 3.12).

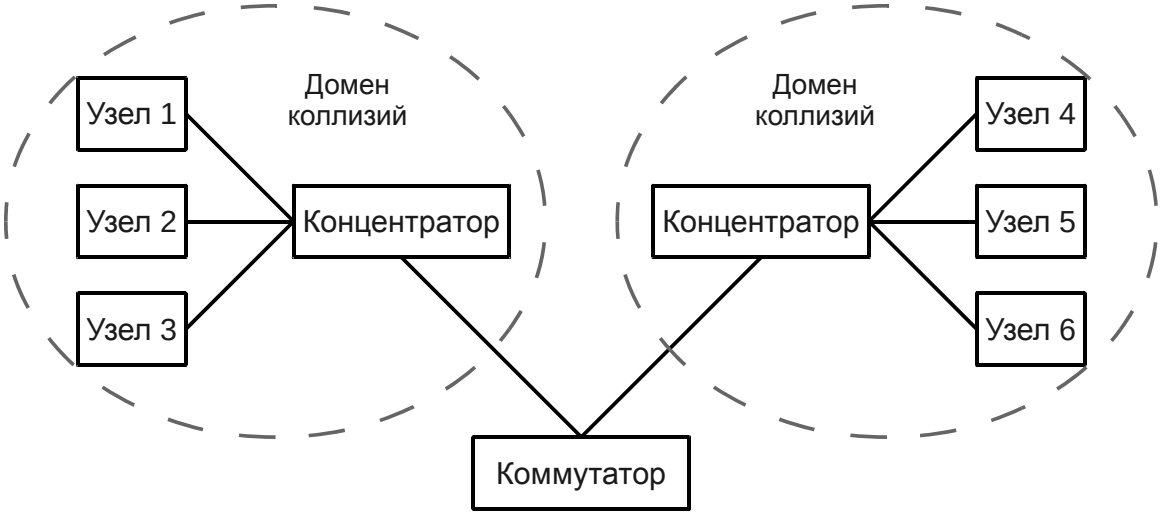


Рис. 3.12. Топология сети с концентраторами и коммутатором [7]

Концентратор (Hub) работает на физическом уровне. Двоичные сигналы, приходящие на любой порт концентратора, воспроизводятся на остальных портах. Таким образом, все узлы, подключённые к концентратору, входят в один *домен коллизий*.

В отличие от концентратора, коммутатор (Switch) работает на канальном уровне. Кадры передаются только на тот интерфейс (разъём для подключения витой пары, оптоволоконной линии, аудиоантеннны), к которому подключен узел назначения. Таким образом домены коллизий локализуются. Коммутатор формирует очередь кадров на каждом интерфейсе и использует протокол CSMA. Наличие буферов на интерфейсах коммутатора позволяет согласовывать сегменты сети, использующие разные скорости передачи данных и разные сетевые технологии физического уровня.

Коммутатор (Switch) и мост (Bridge) являются очень близкими устройствами. Мост использует центральный процессор для обработки поступающих кадров, а в коммутаторе применяется коммутационная матрица. На сегодняшний день мосты практически не используются. Коммутаторы и мосты имеют большее время задержки распространения кадров, чем концентраторы, так как они анализируют кадры.

Коммутатор является самообучающимся устройством. Для того, чтобы определять, на какой интерфейс следует передать очередной

кадр, коммутатор формирует специальную таблицу, в которую записываются номер интерфейса, MAC-адрес подключенного к интерфейсу адаптера, а также время (табл. 3.2).

Таблица 3.2

Фрагмент таблицы коммутатора

MAC-адрес	Интерфейс коммутатора	Время
00:24:1D:84:41:62	1	08:05:00
00:0E:A6:21:12:4D	1	08:35:03
00:11:1B:78:B4:12	3	08:35:04

В момент включения коммутатора таблица пуста. Все приходящие кадры передаются на все интерфейсы. По мере поступления кадров на очередной интерфейс коммутатор считывает адрес узла-отправителя и заполняет таблицу. Каждая запись имеет конечное время жизни. Если тот или иной адаптер в течение длительного периода не обнаруживает своей активности, запись из таблицы исключается. Таким образом таблица поддерживается в актуальном состоянии. Необходимо обратить внимание на то, что широковещательные пакеты передаются на все интерфейсы. Коммутатор имеет ограниченные возможности защиты от *«широковещательного шторма»*.

Коммутатор осуществляет функции *фильтрации* и *продвижения* кадров. Пусть на интерфейс 1 коммутатора (см. табл. 3.2) приходит кадр, адресованный узлу 00:0E:A6:21:12:4D. Коммутатор отыскивает соответствующий адрес в таблице и определяет, что кадр надо игнорировать, таким образом выполняется фильтрация. Пусть на интерфейс 1 приходит кадр, адресованный узлу 00:11:1B:78:B4:12, такой кадр будет передан на интерфейс 3, выполняется продвижение.

Таблицы маршрутизации часто создаются вручную. Однако иерархическая система адресов позволяет ограничивать распространение широковещательных пакетов. Гибкая настройка маршрутизации также позволяет оптимизировать передачу трафика по нескольким каналам (об этом – в гл. 4).

Структура сети (см. рис. 3.12) имеет тот недостаток, что при выходе из строя коммутатора сеть распадётся на несвязанные сегменты. Для повышения надёжности необходимо предусмотреть резервные пути, включив ещё несколько коммутаторов (рис. 3.13).

В такой топологии возникает проблема зацикливания и размножения кадров. Для предотвращения этих эффектов используется протокол *STP – Spanning Tree Protocol IEEE 802.1D (протокол связующего дерева)*. Коммутаторы обмениваются друг с другом специальными кадрами BPDU – Bridge Protocol Data Units и виртуально отключают некоторые интерфейсы так, чтобы исключить циклические пути.

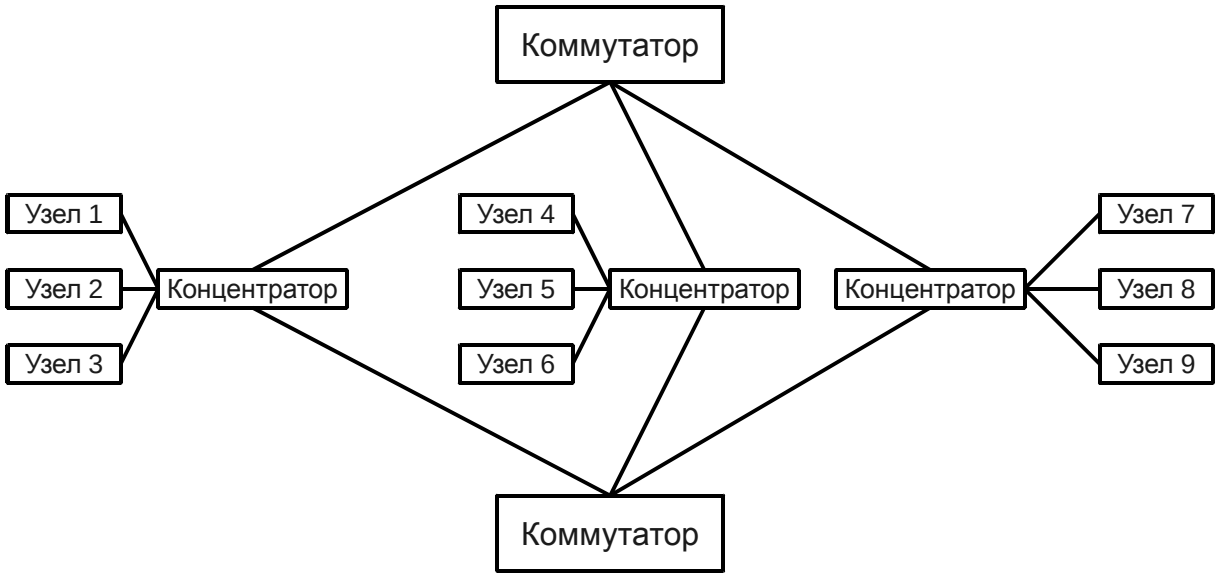


Рис. 3.13. Топология сети с резервными путями [3, 7]

Алгоритм построения связующего дерева включает следующие действия:

- выбирается *корневой коммутатор*. Корневым становится устройство с наименьшим BrigeID (приоритет и MAC-адрес коммутатора);
- прочие коммутаторы находят кратчайший путь до корневого и соответствующий интерфейс назначают *корневым*;
- для каждого сегмента сети определяется кратчайший путь до корневого интерфейса, соответствующие интерфейсы становятся *назначенными*;
- все интерфейсы, кроме корневых и назначенных, отключаются;
- рассылка кадров BDPU выполняется с определённой периодичностью (по умолчанию, две секунды), при необходимости строится новое связующее дерево.

На основе коммутаторов можно строить *виртуальные сети* (VLAN – Virtual LAN). Виртуальной сетью называется группа узлов,

трафик которых изолирован от трафика других узлов. Организация нескольких виртуальных сетей позволяет гибко управлять правами доступа к ресурсам сети и изолировать широковещательные шторма. Виртуальные сети могут пересекаться (например, несколько групп узлов могут иметь доступ к одному серверу). Виртуальные сети объединяются при помощи маршрутизаторов или коммутаторов третьего уровня.

3.8. Помехоустойчивое кодирование на канальном уровне

На сегодняшний день большинство средств передачи и хранения данных используют тот или иной метод помехоустойчивого кодирования. Широко распространённым вариантом является *циклический избыточный код* (CRC – Cyclic Redundancy Code) [1, 5, 8].

Поле CRC (рис. 3.14) содержится в каждом кадре данных и позволяет с высокой достоверностью выявлять ошибки, возникающие при передаче кадра через физическую среду. Поле CRC обеспечивает достоверную передачу информации только в пределах физического сегмента сети.

Рис. 3.14. Данные и циклический избыточный код

Адаптер, обрабатывающий предназначенные для него кадры, проверяет их корректность. Если ошибок не обнаружено, пакет данных, содержащийся в кадре, передаётся на сетевой уровень. В том случае, когда происходит дальнейшая транспортировка этих пакетов (например, маршрутизатором), формируется новый кадр и вычисляется новый избыточный код.

Если адаптер обнаруживает ошибку, кадр может быть просто отброшен. Таким образом, задача надёжной доставки данных от узла-отправителя до узла-получателя в пределах одного физического сегмента сети, и тем более, в нескольких сетях, выполняется протоколами сетевого уровня.

Коротко рассмотрим основы построения циклического избыточного кода.

Пусть $D(x)$ – полином, соответствующий исходным данным, N – количество разрядов CRC. Полином $G(x)$ степени N будем называть *порождающим*. Циклический избыточный код $R(x)$ находится как остаток от деления $D(x)$ на $G(x)$:

$$R(x) = D(x)x^N \cdot mod\ G(x), \qquad (3.1)$$

где $D(x)x^N$ – исходное сообщение, к которому дописаны N нулей в конце;

mod – операция вычисления остатка от деления полиномов.

Корректность полученного кадра можно проверить, разделив последовательность $D(x)$ $R(x)$ на порождающий полином $G(x)$. В случае отсутствия ошибок остаток должен быть равен нулю.

В разных стандартах используются разные порождающие полиномы, что вносит некоторую путаницу. Возможна ситуация, когда оборудование, произведённое разными фирмами, не способно взаимодействовать друг с другом из-за того, что используются отличающиеся алгоритмы вычисления CRC в служебных кадрах.

Группа стандартов IEEE 802.3 использует порождающий полином $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$.

Такой циклический избыточный код позволяет обнаруживать однократные ошибки (ошибки в одном бите кадра), двукратные и трёхкратные ошибки для кадров размером до 1500 байт, а также все ошибки с нечётным числом ошибочных битов.

3.9. Протоколы ARP, RARP

Связь физических адресов (MAC-адресов) сетевых адаптеров и IP-адресов узлов сети обеспечивает протокол ARP (Address Resolution Protocol) – протокол разрешения адресов. Каждый узел ведёт таблицу соответствия MAC-адреса сетевому адресу [3, 7]. Пример записей в памяти узла в табл. 3.3.

Таблица 3.3

Фрагмент таблицы ARP-модуля

IP-адрес	MAC-адрес	Время
10.7.21.1	00:24:1D:84:41:62	08:05:00
10.7.125.293	00:0E:A6:21:12:4D	08:35:03
10.7.212.91	00:11:1B:78:B4:12	08:35:04

Пусть узлу с IP-адресом 10.7.121.121 требуется передать данные на узел 10.7.121.122. Если адрес получателя уже есть в таблице, то соответствующие данные передаются сетевому адаптеру, который формируется кадр с MAC-адресами получателя и отправителя. В этот кадр инкапсулируется пересылаемый пакет.

Если нужный IP-адрес в таблице не найден, то узел посылает ARP-запрос. В кадре в качестве MAC-адреса получателя указывается широковещательный адрес FF:FF:FF:FF:FF:FF, тип кадра указывает на протокол ARP, пакет данных содержит IP-адрес отправителя и IP-адрес получателя (рис. 3.15).

Адрес получателя: FF:FF:FF:FF:FF:FF	Адрес источника: 74:F0:6D:07:80:36	Протокол: ARP	Пакет данных: цель 10.7.121.122, источник 10.7.121.121	CRC

запрос «Кто имеет IP 10.7.121.122? Спрашивает узел 74:F0:6D:07:80:36»

Адрес получателя: 74:F0:6D:07:80:36	Адрес источника: 00:24:1D:84:41:62	Протокол: ARP	Пакет данных: цель 10.7.121.121, источник 10.7.121.122	CRC

ответ «Мой IP 10.7.121.122. Отвечает узел 00:24:1D:84:41:62»

Рис. 3.15. Кадры с запросом и ответом по протоколу ARP

Все адаптеры локальной сети обрабатывают широковещательный запрос и передают его на сетевой уровень. Если один из узлов обнаруживает, что запрошен его MAC-адрес, формируется ответный пакет, аналогичный полученному, только вместо широковещательного MAC-адреса указывается адрес узла-получателя.

Узлы обрабатывают все ARP-запросы, формируя таблицу адресов. Записи в таблице имеют конечное время жизни (как правило, 20 мин.), таким образом таблица поддерживается в актуальном состоянии.

Существует также обратный протокол RARP (Reverse ARP), позволяющий определить IP-адрес по известному MAC-адресу.

3.10. Семейство протоколов PPP

PPP (Point to Point Protocol) – протокол передачи от точки к точке [3, 7]. Семейство PPP включает множество стандартов. Инициализацию и завершение сеанса связи выполняет протокол LCP (Link Control Protocol) – протокол управления каналом. Диаграмма, иллюстрирующая фазы работы протокола PPP, показана на рис. 3.16.

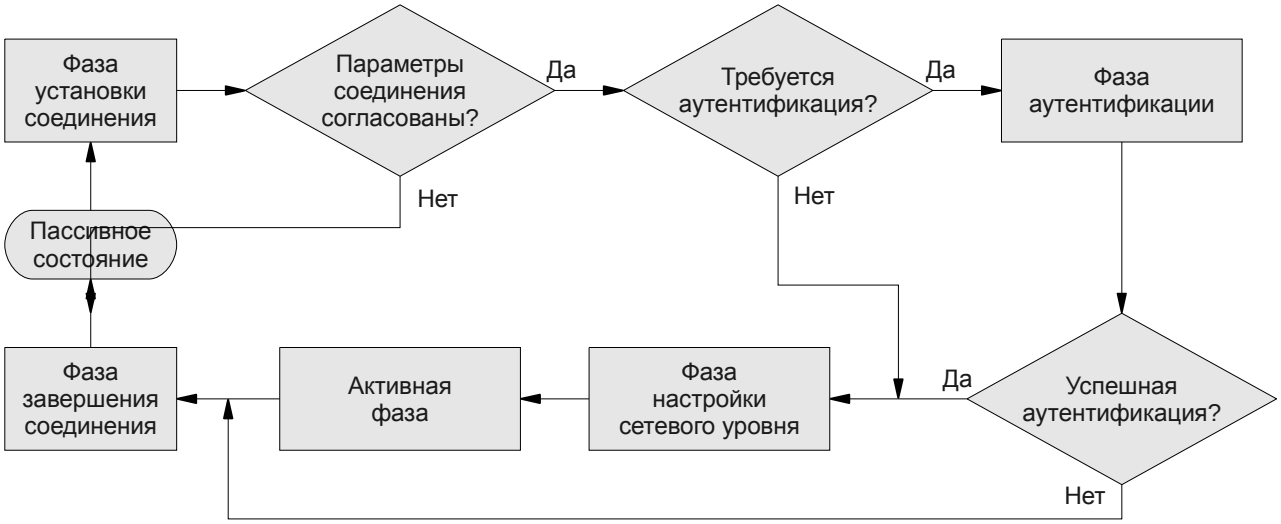


Рис. 3.16. Фазы работы протокола PPP

Так как протокол PPP рассчитан на связь только двух узлов, то можно использовать физически раздельные линии (например, различные витые пары) для передачи в двух направлениях. Таким образом коллизии полностью исключены и возможно дуплексное соединение.

Формат кадра протокола PPP представлен на рис. 3.17. Кадр начинается и заканчивается байтом флага 01111110 (0x7E). Поле адреса может содержать только одно значение 11111111 (0xFF), управляющее поле – 00000011 (0x03). Поле протокола указывает на протокол, инкапсулированный в поле данных. Поле CRC служит для обнаружения ошибок передачи. Так как поле адреса и управляющие поля не несут информации, в процессе установки соединения может быть указан формат кадра без этих полей.

Флаг, 1 байт 0x7E	Адрес, 1 байт 0xFF	Управление, 1 байт 0x03	Протокол, 2 байта	Данные, переменная длина	CRC, 2 - 4 байта	Флаг, 1 байт 0x7E
Кадр протокола PPP						

Рис. 3.17. Формат кадра протокола PPP

Если байт 01111110 (0x7E) встречается в поле данных, то заменяется последовательностью 01111101 01011110 (0x7D 0x5E). Если в поле данных встречается байт 01111101 (0x7D), то заменяется на 01111101 01011101 (0x7D 0x5D). Помимо этого, все байты со значениями меньше, чем 0010000 (0x20) также подменяются двухбайтовыми последовательностями. В частности, байт 00000001 (0x01) передаётся как 01111101 00100001 (0x7D 0x21). Протокол PPP используется для модемных соединений, а модемы могут воспринять байты со значениями от 00000000 (0x00) до 00011111 (0x1F) как управляющие последовательности.

Интернет-провайдерами России для предоставления доступа к Всемирной сети широко используются протоколы PPPoE и PPTP. Протокол PPPoE (Point-to-point protocol over Ethernet) – протокол передачи от точки к точке через Ethernet выполняет инкапсуляцию кадров PPP в кадры Ethernet. PPPoE также предоставляет услуги аутентификации, шифрования, сжатия данных. Протокол PPTP (Point-to-Point Tunneling Protocol) – протокол передачи от точки к точке с использованием тоннеля выполняет инкапсуляцию кадров PPP в IP-пакеты.

Одним из протоколов аутентификации является CHAP (Challenge Handshake Authentication Protocol) – протокол аутентификации с запросом и рукопожатием. Протокол CHAP используется для периодической проверки аутентичности узла с использованием трёхшаговой процедуры. Аутентификация выполняется после фазы установки соединения и состоит из следующих шагов:

- сервер посылает запрос клиенту;
- клиент, используя данные в запросе и пароль, вычисляет хеш-функцию, значение которой передаёт серверу;
- сервер вычисляет хеш-функцию и сравнивает значение с полученным от клиента. Если значения совпадают, то аутентификация считается успешной, в противном случае соединение должно быть завершено.

Через случайный интервал времени сервер может повторить запрос аутентификации. При помощи протокола CHAP возможна также взаимная аутентификация двух узлов. Вначале первый узел посылает запрос второму, после завершения процедуры второй узел посылает запрос первому.

Недостатком протокола CHAP является необходимость хранить пароль в открытом виде на каждом из узлов.

Выводы по главе 3

Канальный уровень модели OSI:

- обеспечивает доставку кадров с инкапсулированными пакетами данных в пределах физического сегмента сети от узла к узлу;
- решает задачу разделения физической среды передачи данных между узлами, подключенными к этой среде;
- обеспечивает локальную (в пределах сегмента сети) целостность кадров;
- в ряде случаев решает задачу аутентификации узлов.

В табл. 3.4 приведены некоторые широко используемые стандарты Ethernet и Wi-Fi, в которые входят спецификации канального уровня.

Таблица 3.4

Некоторые стандарты Ethernet и Wi-Fi

Стандарт	Физическая среда	Скорость	Максимальное расстояние между узлами	Максимальное число узлов в сегменте	Максимальная длина сети
1	2	3	4	5	6
10BASE-5	Коаксиальный кабель, толщина 9 мм, Z = 50 Ом	10 Мбит/с	500 м	100	2,5 км
10BASE-2	Коаксиальный кабель, толщина 5 мм, Z = 50 Ом	10 Мбит/с	185 м	30	925 м
10BASE-T	Витая пара категории 3	10 Мбит/с	100 м	1024	500 м
10BASE-F	Оптический кабель	10 Мбит/с	До 2 км	2	
100BASE-TX	Витая пара категории 5	100 Мбит/с	100 м	1024	205 м
100BASE-FX	Оптический кабель (многомодовый)	100 Мбит/с	До 2 км	1024	
100BASE-LX10	Оптический кабель (одномодовый)	100 Мбит/с	До 10 км	2	
1000BASE-T	Витая пара категории 5	1 Гбит/с	100 м	2	
1000BASE-SX	Оптический кабель (многомодовый)	1 Гбит/с	До 550 м	2	
1000BASE-LX	Оптический кабель (одномодовый)	1 Гбит/с	До 5 км	2	

1000BASE-LH	Оптический кабель (одномодовый)	1 Гбит/с	До 100 км	2	
10GBASE-T	Витая пара категории 6	10 Гбит/с	100 м	2	
10GBASE-SR	Оптический кабель (многомодовый)	10 Гбит/с	До 300 м	2	

Окончание табл. 3.4

1	2	3	4	5	6
10GBASE-LR	Оптический кабель (одномодовый)	10 Гбит/с	До 25 км	2	
100GBASE-LR4	Оптический кабель (одномодовый)	100 Гбит/с	До 10 км	2	
IEEE 802.11b	Радиочастоты в области 2.4 ГГц	До 11 Мбит/с	До 30 м	-	До 8 км (направленные антенны)
IEEE 802.11g	Радиочастоты в области 2.4 ГГц	До 20 Мбит/с	До 30 м	-	До 8 км (направленные антенны)
EEE 802.11n	Радиочастоты в области 2.4 ГГц и в области 5 ГГц	До 300 Мбит/с			

Вопросы для самопроверки

1. Опишите особенности технологии CSMA.
2. Пусть задержка распространения сигнала в канале связи между двумя узлами составляет τ. Пропускная способность канала связи – R. Два узла одновременно начинают передачу кадра длительностью L. Произойдёт ли коллизия, если $\tau < L/R$? Объясните, почему.
3. Зачем в начале кадра Ethernet передаётся преамбула?
4. Почему ARP-запросы посылаются в широковещательном кадре? Почему ARP-ответы посылаются в кадре с MAC-адресом получателя?
5. Многие функции сетевого адаптера могут выполняться программно центральным процессором узла. Каковы преимущества и недостатки передачи этих функций от адаптера к узлу?
6. Как обеспечивается уникальность MAC-адресов?
7. В чём заключается процедура экспоненциального отката в технологии CSMA? Для чего эта процедура нужна?

8. Рассмотрите структуру локальной сети в вашем компьютерном классе. Определите размер домена коллизий.

9. Какие функции выполняет кадр явного подтверждения ACK в сетях с радиодоступом?

10. Каковы функции коммутатора в сети Ethernet?

4. СЕТЕВОЙ УРОВЕНЬ

Задача технологий сетевого уровня – разработка маршрутов доставки пакетов от отправителей к получателям [1, 3, 7]. Причём каждому пакету, направляющемуся по своему адресу, часто требуется преодолевать несколько различных транзитных участков между маршрутизаторами. Функции сетевого уровня более сложные, чем функции уровня передачи данных.

На сетевом уровне сосредоточивается служебная информация о топологии подсети связи (в т. ч. множестве всех маршрутизаторов и их взаимосвязях). Задача сетевого уровня – оптимальный выбор нужного пути по подсети, обеспечивая, по возможности, равномерную нагрузку на маршрутизаторы. Если отправитель и получатель находятся в различных сетях, то на сетевом уровне решаются проблемы их корректного сопряжения.

4.1. Коммутация пакетов с ожиданием

Сетевой уровень обычно реализуется в рамках определённой физической топологии (рис. 4.1) [3, 7]. Основными компонентами схемы являются устройства ввода-вывода информации пользователей и устройства связи и коммутации оператора сети. Последние представлены
маршрутизаторами, соединёнными линиями связи.

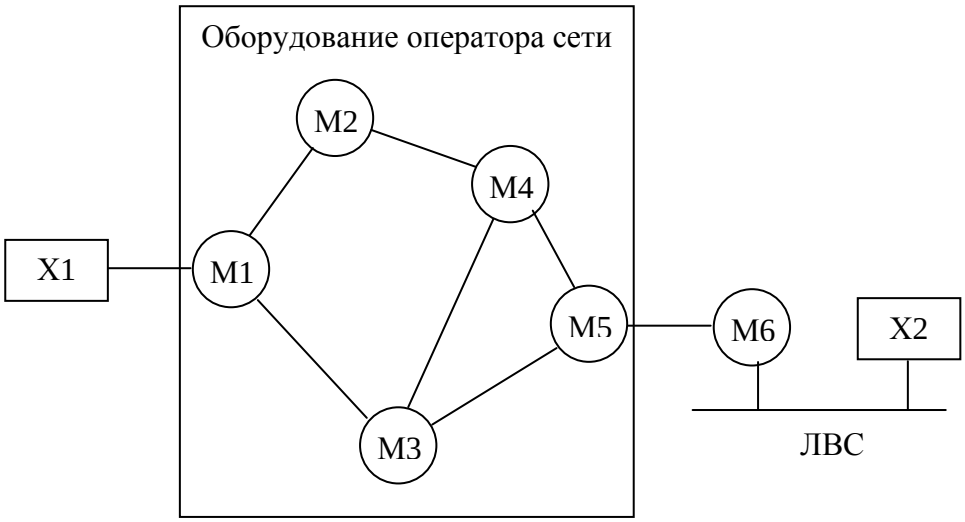


Рис. 4.1. Физическая среда функционирования сетевого уровня

Хост X1 соединён по выделенной линии с маршрутизатором M1 оператора сети. Хост X2 находится в ЛВС пользователя с маршрутизатором M6, который связан с сетью по выделенной линии. По предназначению и протоколам функционирования все маршрутизаторы идентичны.

Хост передаёт пакет на ближайший Маршрутизатор своей ЛВС или непосредственно по соединению оператора сети. Пакет записывается в буфер обмена и хранится до тех пор, пока не будет принят полностью, включая контрольную сумму. Затем он передаётся по цепочке маршрутизаторов до пункта назначения. Описанный механизм называется коммутацией пакетов с ожиданием.

4.2. Сервисы, предоставляемые транспортному уровню

Сетевой уровень предоставляет транспортному уровню сервисы в виде интерфейсов между ними. При этом должны выполняться следующие требования [3, 7]:

1) сервисы сетевого уровня не должны зависеть от технологии маршрутизатора;

2) транспортный уровень должен быть независимым от количества, типа и топологии подсетей с маршрутизаторами;

3) сетевые адреса, доступные транспортному уровню, должны использовать единую систему нумерации в ЛВС и глобальных сетях.

При разработке сети нет чёткого правила в написании детальной спецификации сервисов, которые обычно должны предоставляться транспортному уровню. В сетевых технологиях существуют противоположные точки зрения по вопросу о том, какие сервисы должен предоставлять сетевой уровень – ориентированные на соединение или не требующие соединений.

Разработчики Интернет-сообщества считают, что работа маршрутизатора заключается только в перемещении пакетов и никакие иные функции ему не нужны. Подсеть обладает априорной ненадёжностью независимо от того, как она спроектирована. Хосты должны это учитывать, обнаруживая и исправляя ошибки своими силами, а также самостоятельно управлять потоком. Поэтому сетевой сервис не должен требовать установки соединения и состоять в основном из примитивов SEND PACKET (передача пакета) и RECEME PACKET (приём пакета). Сюда нельзя включать упорядочивание пакетов и контроль потока – всё равно эти действия будет выполнять хост. От выполнения одной и той же работы дважды качество обслуживания не повысится. Кроме того, каждый пакет должен содержать полный адрес получателя, так как пересылка производится независимо от предыдущих пакетов.

Напротив, разработчики телефонных компаний полагают, что сеть должна предоставлять надёжный сервис с соединением. С точки зрения управления телефонными системами, качество обслуживания – определяющий фактор, а без установления соединения в подсети трудно добиться приемлемых результатов, особенно когда дело касается трафика реального масштаба времени – например, передачи голоса и видео.

Примеры технологий, реализующих эти крайние позиции – это Интернет и ATM. Интернет всегда предоставляет не требующие установления соединения сервисы сетевого уровня, а система ATM – ориентированные на соединения.

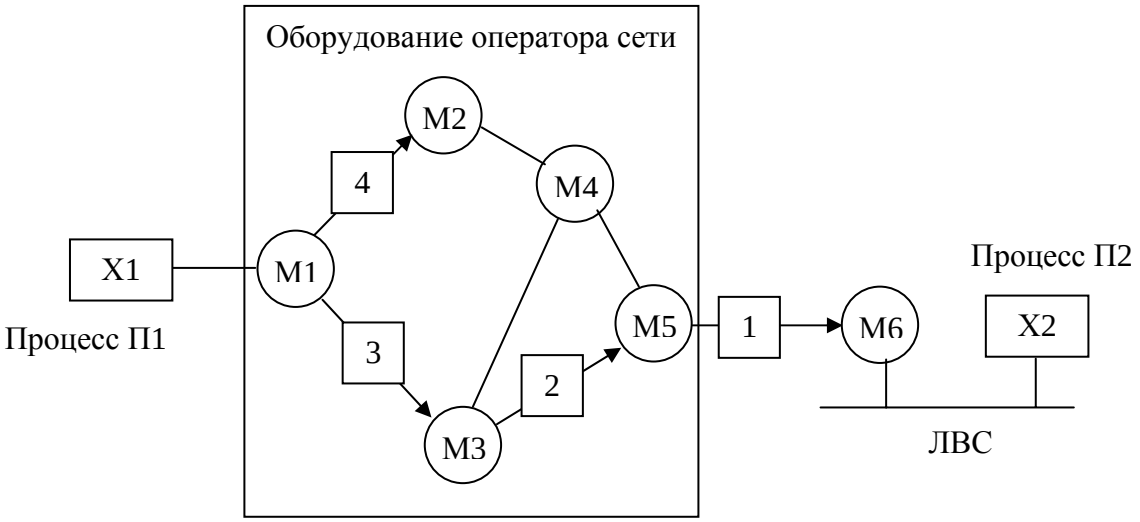
4.3. Сервис без установления соединения

Сервис без установления соединения предполагает, что пакеты поступают в подсеть по отдельности и их маршруты рассчитываются независимо. Тогда никакой предварительной настройки не требуется.

В этом случае пакеты называются дейтаграммами, а подсети – дейтаграммными [3]. При использовании сервиса с соединением путь между маршрутизаторами отправителя и получателя должен быть установлен до начала транспортировки пакетов. Такое соединение называется виртуальным каналом, а подсеть - подсетью виртуального канала.

Рассмотрим принцип работы дейтаграммной подсети (рис. 4.2) [3]. Пусть процесс П1 посылает длинное сообщение для П2. Он передаёт информацию транспортному уровню, сообщает ему, что требуется доставить данные процессу П2, выполняющемуся на хосте Х2. Код транспортного уровня исполняется на хосте Х1. Обычно он является частью операционной системы. Заголовок транспортного уровня вставляется в начало сообщения. В таком виде оно передаётся на сетевой уровень. Обычно это ещё одна процедура операционной системы.

Предположим, что сообщение в 4 раза длиннее максимального размера пакета. Тогда сетевой уровень должен разбить его на 4 пакета и послать их поочерёдно на маршрутизатор М1 с использованием протокола двухточечного соединения, например, PPP. Начинает работать оператор сети. Каждый маршрутизатор имеет свою внутреннюю таблицу, по которой он определяет дальнейший путь пакета при каждом из возможных адресов назначения. Каждая запись таблицы состоит из двух полей: пункта назначения (адресата) и выходящей линии для данного адресата. Во втором поле могут использоваться только линии, непосредственно соединённые с данным маршрутизатором. Например, на рис. 4.2 у маршрутизатора М1 имеется только две исходящие линии – к М2 и М3. Поэтому все входящие пакеты должны пересылаться на один из этих двух маршрутизаторов, даже если они не являются адресатами.



В начале

Назначение	Линия
M1	-
M2	M2
M3	M3
M4	M2
M5	M3
M6	M3

В конце

M2	M2
M3	M3
M4	M2
M5	M2
M6	M3

Таблица маршрутизатора M3

M1	M4
M2	M1
M3	-
M4	M4
M5	M2
M6	M5

Таблица маршрутизатора M5

M1	M4
M2	M4
M3	M3
M4	M4
M5	-
M6	M6

Рис. 4.2. Маршрутизация в дейтаграммной подсети

Пакеты 1, 2 и 3, прибывая по сети на маршрутизатор M1, кратковременно сохраняются в буферной памяти для проверки их корректного приёма по контрольной сумме. Затем, по таблице M1 они передаются на маршрутизатор M3. Далее пакет 1 уходит на M5, откуда поступает на маршрутизатор ЛВС M6. После прибытия на M6 пакет инкапсулируется в кадр уровня передачи данных и передаётся на хост X2 по ЛВС. Пакеты 2 и 3 следуют по тому же маршруту. Пакет 4 после прибытия на M1 передаётся на маршрутизатор M2, несмотря на то, что адресом назначения является M6, как у первых трёх пакетов. По различным причинам маршрутизатор M1 может послать пакет 4 по новому маршруту. Причиной может быть затор на линии M1-M3-M5, возникший при пересылке первых трёх пакетов. Такие явления возникают при ограниченной пропускной способности каналов связи или возникновении внешних помех на линиях. В результате маршрутизатор M1 обновляет свою таблицу (см. рис. 4.2 под надписью «В кон-

це»). Решение о корректировке маршрута принимается алгоритмом маршрутизации.

4.4. Сервис с установлением соединения

Идея состоит в предотвращении выбора своего маршрута для каждого пакета (см. рис. 4.2) [3]. Вместо этого устанавливается соединение, маршрут от источника сообщения до получателя прописывается в настройках системы и хранится в специальных таблицах, встроенных в маршрутизаторы. Один и тот же маршрут используется для всего трафика, проходящего через данное соединение. Так работает телефонная система. Когда соединение разрывается, виртуальный канал также прекращается своё существование. При использовании сервиса с установлением соединения каждый пакет включает идентификатор виртуального канала (рис. 4.3).

Хост X1 устанавливает соединение с хостом X2. Это соединение запоминается и становится первой записью во всех таблицах маршрутизации. Первая строчка таблицы маршрутизатора M1 означает, что если пакет с идентификатором соединения 1 пришёл с хоста X1, то его надо направить на M3 с тем же идентификатором.

Если хост X3 захочет установить соединение с X2, то он выбирает идентификатор соединения 1 (в данном случае, у него нет выбора, это единственное существующее соединение) и просит подсеть установить виртуальный канал. Так в таблице появляется вторая запись. Здесь возникает конфликт, потому что если M1 может отличить пакеты соединения 1, пришедшие от X1, от пакетов соединения 1, пришедших с X3, то M3 такой возможности не имеет. Поэтому M1 присваивает новый идентификатор соединения исходящему трафику и тем самым создаёт второе соединение. Маршрутизаторам нужна возможность изменения идентификаторов соединения в исходящих пакетах для предотвращения подобных конфликтов. Иное название процедуры – коммутация меток.

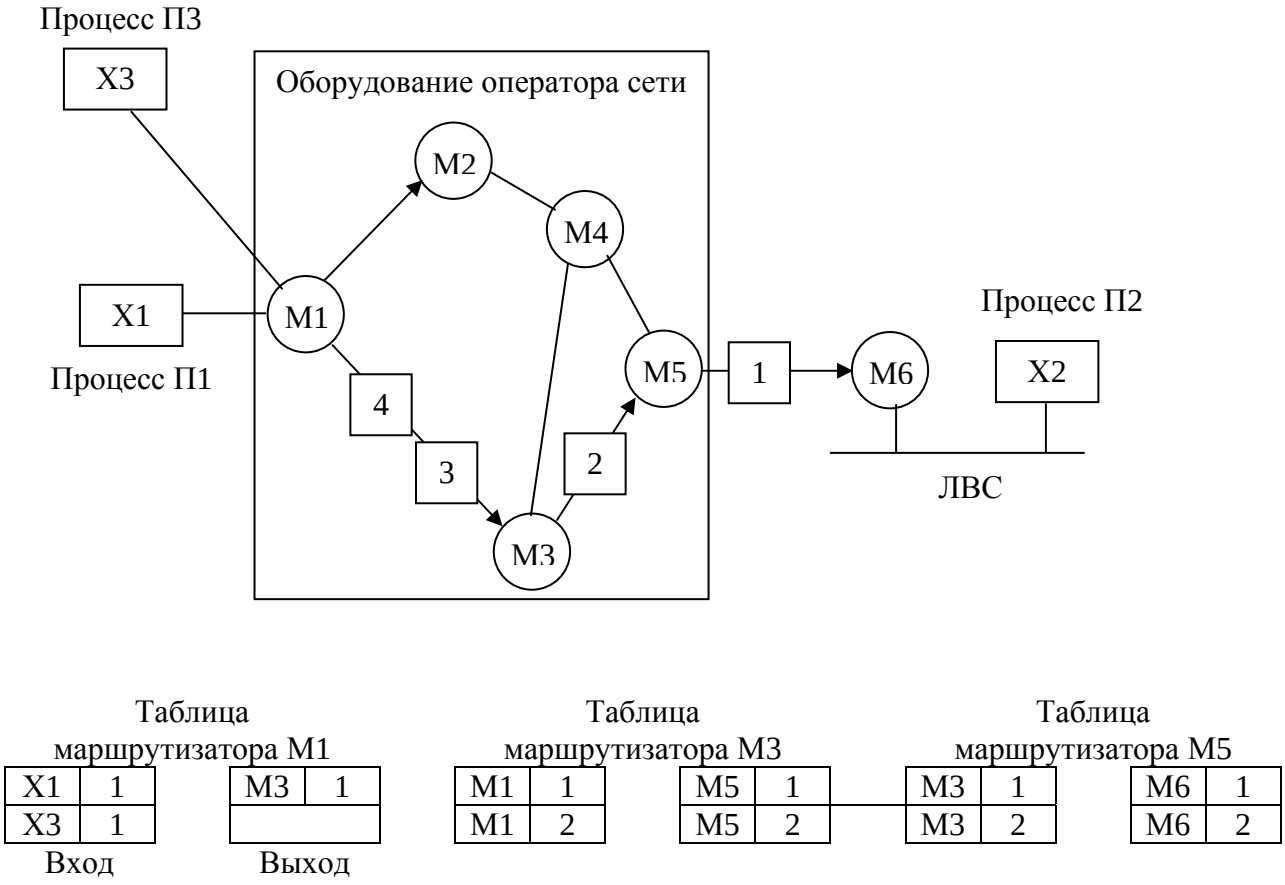


Рис. 4.3. Маршрутизация в подсети виртуального канала

4.5. Сравнение сетей с установлением логических соединений с дейтаграммными сетями

Каждая сеть обладает своими достоинствами и недостатками и используется на практике примерно в равной степени. Основные аргументы сравнения приведены в табл. 4.1 [3].

Оба подхода к созданию сетей реализуются путём компромиссов. Существует компромисс между внутренней памятью маршрутизатора и пропускной способностью. Виртуальные каналы позволяют экономить на адресах получателя, указывая вместо них короткие номера виртуальных каналов. Если размер кадра мал, то полный адрес получателя может составлять довольно существенную часть всего кадра, сильно снижая полезную пропускную способность сети. В то же время, для хранения больших таблиц с адресами в маршрутизаторе может потребоваться много памяти.

Таблица 4.1

Сравнение способа виртуальных каналов с дейтаграммным способом

Признак сравнения	Дейтаграммы	Виртуальные каналы
Установка канала	Не требуется	Требуется
Адресация	Каждый пакет содержит полный адрес отправителя и получателя	Каждый пакет содержит короткий номер виртуального канала
Информация о состоянии	Подсеть не содержит информации о состоянии	Каждый виртуальный канал требует места в таблице подсети
Маршрутизация	Маршрут каждого канала выбирается независимо	Единый маршрут для всех пакетов выбирается при установке виртуального канала
Последствия выхода из строя маршрутизатора	Только потерянные пакеты	Все виртуальные каналы, проходящие через маршрутизатор, прекращают существование
Борьба с перегрузкой	Трудно реализуется	Легко реализуется при наличии достаточного количества буферов для каждого виртуального канала

Второй компромисс при создании сетей – между временем установления соединения и временем обработки адреса. Виртуальный канал требует затрат времени на его установку, но последующая обработка пакетов для маршрутизатора будет проще и быстрее, чем в дейтаграммной сети.

Виртуальные каналы обладают некоторыми преимуществами, помогающими им предоставлять гарантированное качество обслуживания и избегать заторов в подсети, так как ресурсы могут быть зарезервированы заранее, во время установления соединения. Когда начинают прибывать пакеты, необходимая пропускная способность и мощность маршрутизатора будут предоставлены системой. В дейтаграммной подсети предотвращение заторов реализовать значительно сложнее.

В системах обработки транзакций (например, при запросе магазина на верификацию кредитной карты) затраты времени на установление соединения и удаление виртуального канала могут сильно снизить потребительские свойства сети.

Если объём информации, передаваемой во время одного соединения, невелик, то использование виртуального канала не имеет смысла.

В данной ситуации могут оказаться полезными постоянные виртуальные каналы, установленные вручную и не разрываемые месяцами и даже годами.

Недостатком виртуальных каналов является их уязвимость в случае выхода из строя по различным причинам или временного выключения маршрутизатора. Даже если он будет быстро починен и снова включен, все виртуальные каналы, проходившие через него, будут прерваны. Если же в дейтаграммной сети маршрутизатор выйдет из строя, то будут потеряны только те пакеты, которые находились в данный момент на маршрутизаторе (а возможно, лишь некоторые из них, в зависимости от того, были они подтверждены или нет). Обрыв линии связи для виртуальных каналов является фатальным, а в дейтаграммной системе может оказаться почти незамеченным. Кроме того, дейтаграммная система позволяет соблюдать баланс между загрузкой маршрутизаторов и линий связи.

4.6. Алгоритмы маршрутизации

Основная функция сетевого уровня заключается в выборе маршрута для пакетов от начальной до конечной точки. В большинстве сетей пакетам приходится проходить через несколько маршрутизаторов. Единственным исключением здесь являются широковещательные сети, но даже в них маршрутизация является важным вопросом, если отправитель и получатель находятся в разных сетях. Алгоритмы выбора маршрутов и используемые ими структуры данных являются главной целью при проектировании сетевого уровня.

Алгоритм маршрутизации пакетов реализуется той частью программного обеспечения сетевого уровня, которая отвечает за выбор выходной линии для отправки пришедшего пакета. Если подсеть использует дейтаграммную службу, выбор маршрута для каждого пакета должен производиться заново, так как оптимальный маршрут мог измениться. Если подсеть использует виртуальные каналы, маршрут выбирается только при создании нового виртуального канала. После этого все информационные пакеты следуют по выбранному маршруту. Последний случай иначе называется сеансовой маршрутизацией, так как маршрут остаётся в силе на протяжении всего сеанса связи с пользователем.

Есть разница в понятиях маршрутизации, при которой системе приходится делать выбор определённого маршрута следования, и пересылки действием, происходящим при получении пакета. Можно представить себе маршрутизатор как устройство, в котором функционируют два процесса. Один из них обрабатывает приходящие пакеты и выбирает для них по таблице маршрутизации исходящую линию. Такой процесс называется пересылкой. Второй процесс отвечает за заполнение и обновление таблиц маршрутизации. Именно здесь в игру вступает алгоритм маршрутизации.

Вне зависимости от того, отдельно ли выбираются маршруты для каждого пакета или же только один раз – для соединения, желательно, чтобы алгоритм выбора конкретного маршрута обладал определёнными свойствами: корректностью, оптимальностью, надёжностью, устойчивостью, и простотой. Правильность и простота вряд ли требуют комментариев, а вот потребность в надёжности не столь очевидна с первого взгляда. Во время работы большой сети постоянно происходят какие-то отказы аппаратуры и изменения топологии. Алгоритм маршрутизации должен уметь справляться с изменениями топологии и трафика без необходимости прекращения всех задач на всех хостах и перезагрузки сети при каждой поломке маршрутизатора.

Алгоритм сетевой маршрутизации должен также обладать устойчивостью. Существуют алгоритмы выбора маршрута, никогда не приходящие в состояние равновесия, независимо от того, как долго они работают.

Такая цель, как оптимальность, может показаться очевидной, но здесь могут возникнуть определённые сложности. Для примера рассмотрим следующую ситуацию (рис. 4.4). Предположим, что трафик между станциями *A* и *A'*, *B* и *B'*, *C* и *C'* настолько интенсивный, что горизонтальные линии связи оказываются полностью загруженными. Чтобы максимизировать общий поток данных, трафик между станциями *X* и *X'* следовало бы совсем отключить. Однако станции *X* и *X'* тоже должны работать в сети. Очевидно, необходим компромисс между справедливым выделением трафика всем станциям и оптимальным использованием канала в глобальном смысле.

Задача оптимальности решается исходя из конкретно выбранного критерия. Можно попробовать минимизировать среднее время задержки или увеличить общую пропускную способность сети.

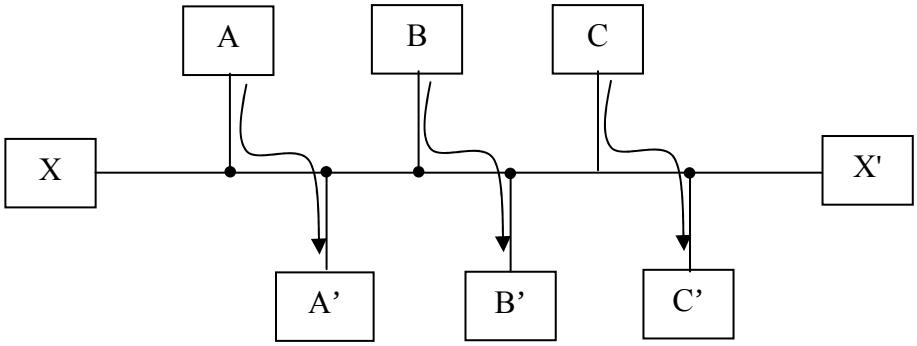


Рис. 4.4. Проблема оптимальности трафика

Однако эти цели противоречат друг другу, поскольку работа любой системы с очередями вблизи максимума производительности предполагает долгое стояние в очередях. В качестве компромисса многие сети стараются минимизировать количество пересылок для каждого пакета, поскольку при этом снижается время прохождения пакета по сети, а также нагрузка на сеть, в результате чего улучшается пропускная способность.

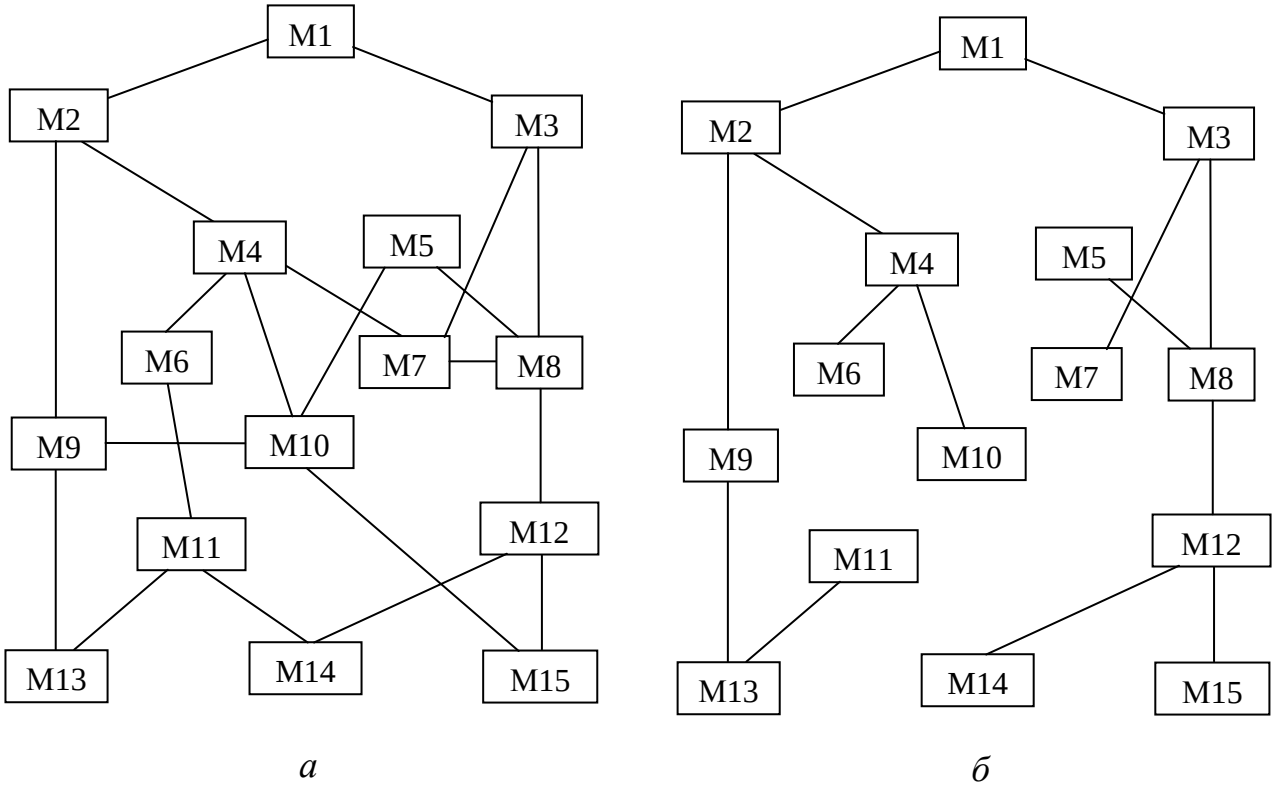
Алгоритмы выбора маршрута можно разбить на два основных класса: адаптивные и неадаптивные [3]. Неадаптивные алгоритмы не учитывают при выборе маршрута топологию и текущее состояние сети и не изменяют трафик на линиях. Вместо этого выбор маршрута для каждой пары станций производится заранее, в автономном режиме. Список маршрутов загружается в маршрутизаторы во время каждой загрузки сети. Такая процедура называется статической маршрутизацией.

Адаптивные алгоритмы, напротив, изменяют решение о выборе маршрутов при изменении топологии, а также, во многих случаях - в зависимости от загруженности линий. Адаптивные алгоритмы отличаются источниками получения информации (такие источники могут быть, например, локальными, если это соседние маршрутизаторы, либо глобальными, если это вообще все маршрутизаторы сети), моментами изменения маршрутов (например, через определённые равные интервалы времени, при изменении нагрузки или при изменении топологии) и данными, использующимися для оптимизации (расстояние, количество транзитных участков или ожидаемое время пересылки).

4.7. Принцип оптимальности маршрута

Основополагающей идеей сетевого уровня является принцип оптимальности [2]. В соответствии с этим принципом, если маршрутизатор *B* располагается на оптимальном маршруте от маршрутизатора *A* к маршрутизатору *C*, то оптимальный маршрут от маршрутизатора *B* к маршрутизатору *C* совпадёт с частью первого маршрута. Чтобы убедиться в этом, обозначим часть маршрута от маршрутизатора *A* к маршрутизатору *B* как *r*1 а остальную часть маршрута – *r*2. Если бы существовал более оптимальный маршрут от маршрутизатора *B* к маршрутизатору *C*, чем *r*2, то его можно было бы объединить с *r*1, чтобы улучшить маршрут от маршрутизатора *A* к маршрутизатору *C*, что противоречит первоначальному утверждению о том, что маршрут *r*1 - *r*2 является оптимальным.

Прямым следствием принципа оптимальности сетей является возможность рассмотрения множества оптимальных маршрутов от всех источников к приёмникам в виде дерева (рис. 4.5). Такое дерево называется входным деревом.



а *б*

Рис. 4.5. Пример топологии сети:
а – подсеть; *б* – входное дерево для маршрутизатора M1

Расстояния между отправителем и получателем сообщений измеряются количеством транзитных участков маршрута. Входное дерево

не обязательно является уникальным. У одной сети может существовать несколько входных деревьев с одинаковыми длинами путей. Цель всех алгоритмов выбора маршрутов заключается в вычислении и использовании входных деревьев для всех маршрутизаторов.

Входное дерево не содержит петель, поэтому каждый пакет будет доставлен за конечное и ограниченное число пересылок. На практике всё это не так просто. Линии связи и маршрутизаторы могут выходить из строя и снова появляться в сети во время выполнения операции, поэтому у разных маршрутизаторов могут оказаться различные представления о текущей топологии сети. Кроме того, ещё не обсуждался вопрос о том, собирает ли маршрутизатор информацию для вычисления входного дерева сам или эта информация поступает к нему каким-то другим образом. Тем не менее принцип оптимальности и входное дерево – это те точки отсчета, относительно которых можно измерять эффективность различных алгоритмов маршрутизации.

4.8. Выбор кратчайшего пути

Изучение алгоритмов выбора маршрутов начинается с широко применяемого в различных формах метода графов благодаря его простоте и понятности. Идея заключается в построении графа подсети, где каждый узел соответствует маршрутизатору, а каждая дуга – линии связи. При выборе маршрута между двумя маршрутизаторами алгоритм просто находит кратчайший путь между ними на графе (рис. 4.6) [2].

Один из способов измерения длины пути состоит в подсчёте количества транзитных участков. В таком случае пути M1-M2-M6 и M1-M2-M4 (рис. 4.6, *а*) имеют одинаковую длину. Но если измерять расстояния в километрах, то окажется, что путь M1-M2-M6 значительно длиннее пути M1-M2-M4.

Кроме количества транзитных участков и физической длины линий возможен учёт и других параметров. Например, каждой дуге графа можно поставить в соответствие среднюю длину очереди и время задержки пересылки, которые определяются каждый час с помощью передачи специального тестового пакета. В таком графе кратчайший путь определяется как самый быстрый путь, а не путь с самой короткой длиной кабеля или путь, состоящий из минимального числа отдельных отрезков кабеля.

В общем случае, параметры дуг графа являются функциями расстояния, пропускной способности, средней загруженности, стоимости связи средней длины очереди, измеренной величины задержки и других факторов. Изменяя весовую функцию, алгоритм может вычислять кратчайший путь с учётом любого количества критериев в различных комбинациях.

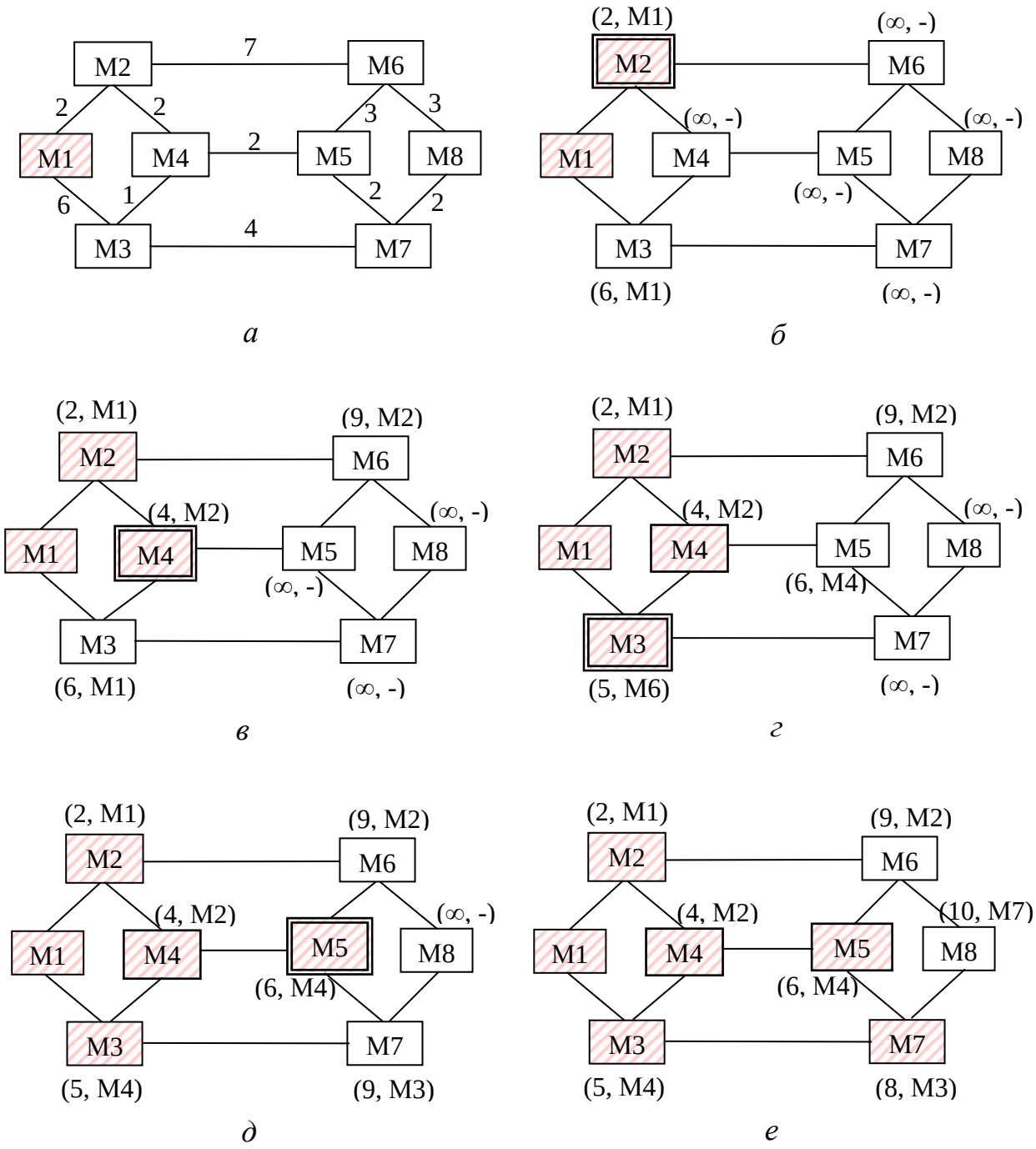


Рис. 4.6. Первые пять шагов вычисления кратчайшего пути от M1 к M8:
а – взвешенный ненаправленный граф; *б* – определение ближайшего узла; *в* – определение текущего рабочего узла; *г-е* – завершающие этапы вычисления пути; 3, 4, 5-й – шаги соответственно

Известно несколько алгоритмов вычисления кратчайшего пути между двумя узлами графа. Один из них был создан Дейкстрой (Dijkstra) в 1959 г. Каждый узел помечается (в скобках) расстоянием до него от узла отправителя по наилучшему известному пути. Вначале пути неизвестны, поэтому все узлы помечаются символом бесконечности. По мере работы алгоритма и нахождения путей отметки узлов изменяются, показывая оптимальные пути. Отметка может быть постоянной или экспериментальной. Вначале все отметки являются ориентировочными. Когда выясняется, что отметка действительно соответствует кратчайшему возможному пути, она становится постоянной и в дальнейшем не изменяется.

Чтобы показать, как работает этот алгоритм, рассмотрим взвешенный ненаправленный граф (см. рис. 4.6, *а*), где весовые коэффициенты соответствуют, например, расстояниям. Нужно найти кратчайший путь от М1 к М8. Для начала розовым пунктиром помечается узел М1 как постоянный. Затем исследуются все соседние с ним узлы, около них указывается расстояние до узла М1. Если отыскивается более короткий путь к какому-либо узлу, то вместе с указанием расстояния в отметке меняется и узел, через который прошёл более короткий путь. Таким образом позже можно восстановить весь путь. Рассмотрев все соседние с М1 узлы, помечается ближайший узел как постоянный (см. рис. 4.6, *б*). Этот узел и становится новым рабочим узлом.

Теперь можно повторить ту же процедуру с узлом М2, исследуя все его соседние узлы. Если сумма расстояния от узла М2 и значения отметки в узле М2 (расстояния от М1 до М2) оказывается меньше, чем отметка у исследуемого узла (уже найденное другим путём расстояние от М1), это значит, что найден более короткий путь, поэтому пометка узла изменяется.

После того, как все соседние с рабочим узлы исследованы и временные отметки при необходимости изменены, по всему графу ищется узел с наименьшей временной отметкой. Этот узел помечается как постоянный и становится текущим рабочим узлом.

На этапе (рис. 4.6, *в*) узел М4 отмечен как постоянный. Предположим, что существует более короткий путь, чем М1-М2-М4, например, М1-М3-М7-М5-М4. В этом случае предполагаемый маршрут исследуется и принимается решение о целесообразности его использования. По пути к М8 все варианты маршрутов просчитываются неоднократно в различных сочетаниях. При этом имеет значение текущее состояние каналов связи – их исправность и скорость передачи данных. Последняя зависит от внешних помех интенсивности трафика.

4.9. Заливка

Метод заливки (лавинной маршрутизации) – это статический алгоритм, при котором каждый приходящий пакет посылается на все исходящие линии, кроме той, по которой он пришёл [2, 3]. Очевидно, что лавинная марштуризация порождает огромное количество дублированных пакетов, даже бесконечное количество в сетях с замкнутыми контурами, если не принять специальных мер. Одна из таких мер состоит в помещении в заголовок пакета счётчика преодолённых им транзитных участков, уменьшаемого при прохождении каждого маршрутизатора. Когда значение этого счётчика падает до нуля, пакет удаляется. В идеальном случае счётчик транзитных участков должен вначале устанавливаться равным длине пути от отправителя до получателя. Если отправитель не знает расстояния до получателя, он может установить значение счётчика равным длине максимального пути в данной подсети.

Альтернативный способ ограничения количества тиражируемых пакетов заключается в их учёте при прохождении через маршрутизатор. Это позволяет не посылать их повторно. Один из методов состоит в том, что каждый маршрутизатор помещает в каждый получаемый от своих хостов пакет порядковый номер. Все маршрутизаторы ведут список маршрутизаторов-источников, в котором сохраняются все порядковые номера пакетов, которые им встречались. Если пакет от данного источника с таким порядковым номером в списке уже есть, то дальше он не распространяется и удаляется.

Чтобы предотвратить неограниченный рост размера списка, можно снабдить все списки счётчиком k, показывающим, что все порядковые номера, вплоть до k, уже встречались. И когда приходит пакет, можно очень легко проверить, не является ли он дубликатом. При положительном ответе такой пакет отвергается. Кроме того, не нужно хранить весь список до k, так как этот счётчик очень действенно подытоживает его.

На практике чаще всего применяется выборочная заливка. В этом алгоритме маршрутизаторы посылают пакеты не по всем линиям, а только по тем, которые идут приблизительно в нужном направлении. Вряд ли есть смысл посылать пакет, направляющийся на запад, по линии, идущей на восток, если только топология сети не представляет собой лабиринт и маршрутизатор не знает об этом.

В большинстве случаев алгоритм лавинной маршрутизации является непрактичным, но, тем не менее, иногда он применяется. Например, в военных приложениях, где большая часть маршрутизаторов в любой момент может оказаться уничтоженной, высокая надёжность алгоритма заливки является, наоборот, желательной. В распределённых базах данных иногда бывает необходимо одновременно обновить все базы данных, и в этом случае заливка также оказывается полезной. Третье применение алгоритма заливки – эталонное тестирование других алгоритмов выбора маршрутов, так как заливка всегда находит все возможные пути в сети, а следовательно, и кратчайшие. Ухудшить эталонные показатели времени доставки могут разве что накладные расходы, вызванные огромным количеством пакетов, формируемых самим алгоритмом заливки.

4.10. Маршрутизация по вектору расстояний

Современные компьютерные сети обычно используют не статические, а динамические алгоритмы маршрутизации, поскольку статические просто не принимают во внимание текущую нагрузку на сеть. Самой большой популярностью пользуются два метода: маршрутизация по вектору расстояний и маршрутизация с учётом состояния каналов [3]. В данном пункте (4.10) изложен первый, в следующем (4.11) будет изложен второй метод. Алгоритмы маршрутизации по вектору расстояний работают, опираясь на таблицы, поддерживаемые всеми маршрутизаторами и содержащие наилучшие известные пути к каждому из возможных адресатов. Для обновления данных этих таблиц производится обмен информацией с соседними маршрутизаторами.

Алгоритм маршрутизации по вектору расстояний иначе называют по именам его создателей: распределённым алгоритмом Беллмана – Форда (Bellman – Ford) и алгоритмом Форда – Фулкерсона (Ford – Fulkerson), (Bellman, 1957; Ford and Filkerson, 1962). Этот алгоритм изначально применялся в сети ARPANET и в Интернете был известен под названием RIP.

При маршрутизации по вектору расстояний таблицы, с которыми работают и которые обновляют маршрутизаторы, содержат записи о каждом маршрутизаторе подсети. Каждая запись состоит из двух частей: предпочитаемого номера линии для данного получателя и предполагаемого расстояния или времени прохождения пакета до

этого получателя. В качестве единиц измерения на практике может использоваться число транзитных участков, миллисекунды, число пакетов, ожидающих в очереди в данном направлении, или ещё что-нибудь подобное.

Предполагается, что маршрутизаторам известно расстояние до каждого из соседей. Если в качестве единицы измерения используется число транзитных участков, то расстояние равно одному транзитному участку. Если же дистанция измеряется временем задержки, то маршрутизатор может измерить его с помощью специального пакета ECHO (эхо), в который получатель помещает время получения и который отправляет обратно как можно быстрее.

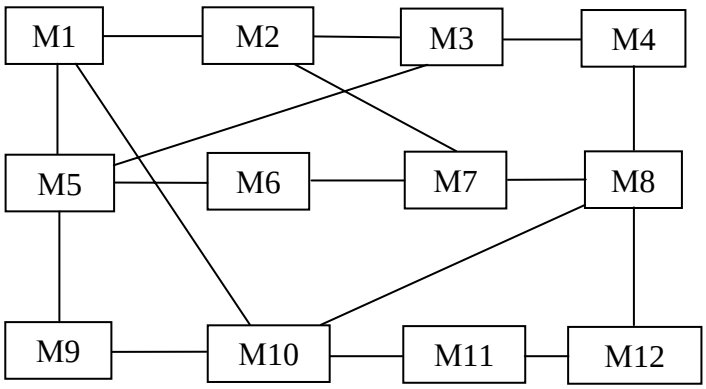
Предположим, что в качестве единицы измерения используется время задержки, и этот параметр относительно каждого из соседей известен маршрутизатору. Через каждые *T* миллисекунды все маршрутизаторы сети посылают своим соседям список с приблизительными задержками для каждого получателя. Они, в свою очередь, также получают подобный список от всех своих соседей. Допустим, одна из таких таблиц пришла от соседа M1, и в ней указывается, что время распространения от маршрутизатора M1 до маршрутизатора M_i равно T_i. Если маршрутизатор знает, что время пересылки до маршрутизатора M1 равно *tn*, тогда задержка при передаче пакета маршрутизатору M_i через маршрутизатор M составит $M_t + m$. Выполнив такие расчёты для всех своих соседей, маршрутизатор может выбрать наилучшие пути и поместить соответствующие записи в новую таблицу. Старая таблица в расчётах не используется.

Процесс обновления таблицы проиллюстрирован на рис. 4.7. Сверху показана подсеть (рис. 4.7, *а*). Первые четыре столбца (рис. 4.7, *б*) показывают векторы задержек, полученные маршрутизатором M10 от своих соседей.

Маршрутизатор M1 считает, что время пересылки от него до маршрутизатора M2 равно 12 мс, 25 мс – до маршрутизатора M3, 40 мс – до M4 и т. д. Предположим, что маршрутизатор измерил или оценил задержки до своих соседей M1, M9, M8 и M11 как 8, 10, 12 и 6 мс соответственно.

Теперь рассмотрим, как M10 рассчитывает свой новый маршрут к маршрутизатору M7. Он знает, что задержка до M1 составляет 8 мс, а M1 думает, что от него до M7 данные дойдут за 18 мс. Таким образом, M10 знает, что если он станет отправлять пакеты для M7 через M1, то задержка составит 26 мс. Аналогично он вычисляет

значения задержек для маршрутов от него до М7, проходящих через остальных его соседей (М9, М8 и М11), и получает соответственно 41 = 31 + 10, 18 = 6 + 12 и 37 = 31 + 6.



а

	M1	M9	M8	M11	Новая расчётная задержка для М10: Т	Линия
M1	0	24	20	21	6	M1
M2	12	36	31	28	20	M1
M3	25	18	19	36	28	M9
M4	40	27	8	24	20	M8
M5	14	7	30	22	17	M9
M6	23	20	19	40	30	M9
M7	18	31	6	31	18	M8
M8	17	20	0	19	12	M8
M9	21	0	14	22	10	M9
M10	9	11	7	10	0	-
M11	24	22	22	0	6	M11
M12	29	33	9	9	15	M12
	Задержка М10-М1 = 8	Задержка М10-М9 = 10	Задержка М10-М8 = 12	Задержка М10-М11 = 6	Новая таблица маршрутов для М10	

Векторы, полученные от четырёх соседей М10

б

Рис. 4.7. Построение таблицы маршрутизации:
а – подсеть; *б* – полученные от М1, М9, М8, М11 векторы и новая таблица маршрутов для М10

Лучшим значением очевидно является 18, поэтому именно оно помещается в таблицу в запись для получателя М7. Вместе

с числом 18 туда же помещается обозначение линии, по которой проходит самый короткий маршрут до M7, то есть M8. Данный метод повторяется для всех остальных адресатов, и при этом получается новая таблица, показанная в виде правого столбца на рис. 4.7, *б*.

4.11. Маршрутизация с учётом состояния линий

Маршрутизация на основе векторов расстояний использовалась в сети ARPANET вплоть до 1979 г., после чего её сменил алгоритм маршрутизации с учётом состояния линий. Отказаться от прежнего алгоритма пришлось по двум причинам. Во-первых, старый алгоритм при выборе пути не учитывал пропускную способность линий. Пока все линии имели пропускную способность 56 кбит/с, в её учёте не было необходимости. Однако стали появляться линии со скоростью 230 кбит/с, а затем и 1,544 Мбит/с, и не принимать во внимание пропускную способность стало невозможно. Конечно, можно было ввести пропускную способность в качестве множителя для единицы измерения, но имелась ещё и другая проблема, заключавшаяся в том, что алгоритм слишком долго приходил к устойчивому состоянию (непредсказуемость событий обрыва и восстановления каналов между маршрутизаторами, а также временных рамок протекания этих процессов и оперативного доведения этой информации до маршрутизаторов). Поэтому старый алгоритм был полностью заменён новым алгоритмом, который называется сейчас маршрутизацией с учётом состояния линий [3]. Варианты этого алгоритма широко применяются в наши дни.

В основе алгоритма лежит простая идея, которую можно изложить в *пяти требованиях к маршрутизатору*. Каждый маршрутизатор должен:

1. Обнаруживать своих соседей и узнавать их сетевые адреса.
2. Измерять задержку или стоимость связи с каждым из своих соседей.
3. Создавать пакет, содержащий всю собранную информацию.
4. Посылать созданный пакет всем маршрутизаторам.
5. Вычислять кратчайший путь ко всем маршрутизаторам.

В результате, каждому маршрутизатору высылаются полная топология и все измеренные значения задержек. После этого для обнару-

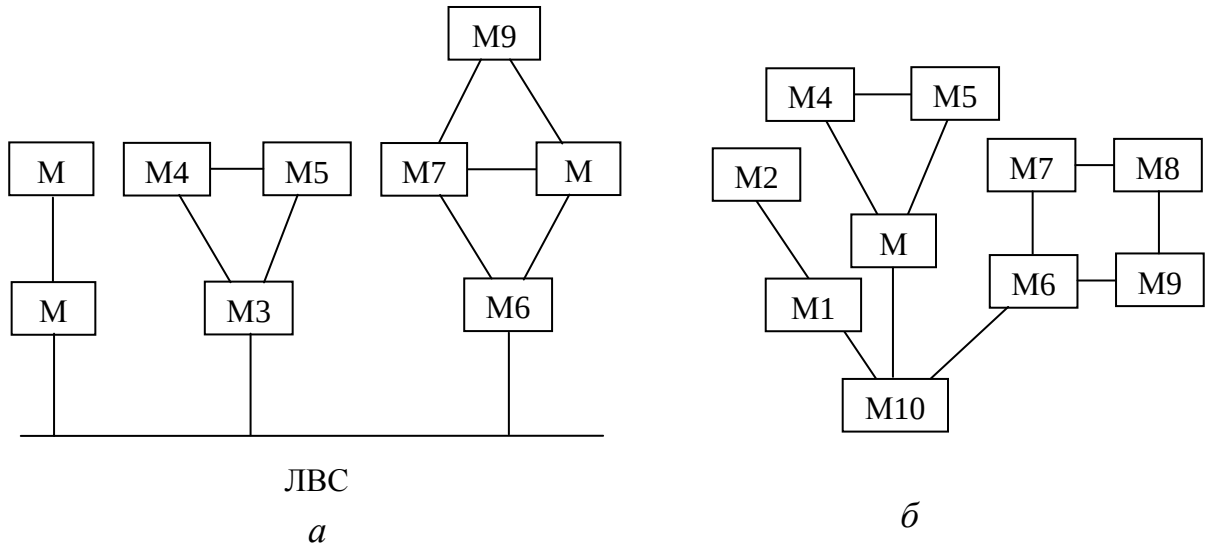
жения кратчайшего пути к каждому маршрутизатору может применяться алгоритм Дейкстры. Далее рассмотрим каждый из этих пяти этапов более подробно.

4.12. Знакомство с соседями (идентификация в сети)

Когда маршрутизатор загружается, его первая задача состоит в получении информации о своих соседях [3]. Он достигает этой цели, посылая специальный пакет HELLO по всем двухточечным линиям. При этом маршрутизатор на другом конце линии должен послать ответ, сообщая информацию о себе. Имена маршрутизаторов должны быть совершенно уникальными, поскольку, если, к примеру, удалённый маршрутизатор слышит, что три маршрутизатора являются соседями маршрутизатора М6, то не должно возникать разночтений по поводу того, один и тот же маршрутизатор М6 имеется в виду или нет.

Когда два или более маршрутизаторов объединены в ЛВС, ситуация несколько усложняется. На рис. 4.8, *а* изображена ЛВС, к которой напрямую подключены три маршрутизатора – М1, М3 и М6. Каждый из них соединён также с одним или несколькими дополнительными маршрутизаторами.

Один из способов моделирования ЛВС состоит в том, что она рассматривается в виде узла графа, как и маршрутизаторы (рис. 4.8, *б*). Сеть изображена в виде искусственного узла М10, с которым соединены маршрутизаторы М1, М3 и М6. Возможность передачи пакетов от М1 к М3 по ЛВС отражается здесь наличием пути М1-М10-М3.



а *б*

Рис. 4.8. Пример системы маршрутизации:
а – маршрутизаторы в ЛВС; *б* – графовая модель этой системы

4.13. Измерение стоимости линии

Алгоритм маршрутизации с учётом состояния линии требует от каждого маршрутизатора знания или хотя бы обоснованной оценки задержки для всех линий связи со своими соседями [3]. Наиболее прямой способ определения этой задержки заключается в посылке по линии специального пакета ECHO, на который другая сторона обязана немедленно ответить. Измерив время двойного оборота этого пакета и разделив его на два, отправитель получает приемлемую оценку задержки. Чтобы получить более точный результат, это действие можно повторить несколько раз, после чего вычислить среднее арифметическое. Конечно, такой метод предполагает, что задержки являются симметричными, что не всегда так.

Возникает закономерный вопрос: надо ли учитывать нагрузку на линию во время измерения задержки? Чтобы учесть загруженность линии, таймер должен включаться при отправке пакета ECHO. Чтобы игнорировать загрузку, таймер следует включать, когда пакет ECHO достигает начала очереди.

Оба способа могут быть аргументированы. Учёт трафика в линии при измерении задержки означает, что если у маршрутизатора есть выбор между двумя линиями с одинаковой пропускной способностью, то маршрут по наименее загруженной линии рассматривается как более короткий. Такой выбор приведет к более сбалансированному использованию линий связи и, следовательно, к более эффективной работе системы.

Рассмотрим подсеть, показанную на рис. 4.9. Она разделена на две части – восточную и западную, которые соединены двумя линиями, M3-M6 и M5-M9.

Предположим, что основная часть потока данных между востоком и западом использует линию M3-M6. В результате эта линия оказывается сильно загруженной и с большими задержками. Учёт времени стояния пакета в очередях при подсчёте кратчайшего пути сделает линию M5-M9 более предпочтительной. После установки новых таблиц маршрутизации большая часть потока данных между востоком и западом переместится на линию M5-M9, и ситуация повторится с точностью до смены одной линии на другую. Аналогично, после ещё одного обновления уже линия M5-M9 окажется в данный момент лучшей. В результате, таблицы маршрутизации будут страдать от не-

затухающих колебаний, что сильно снизит эффективность работы системы. Если же нагрузку не учитывать, то эта проблема не возникнет. Можно поступать по-другому: распределять нагрузку между двумя линиями.

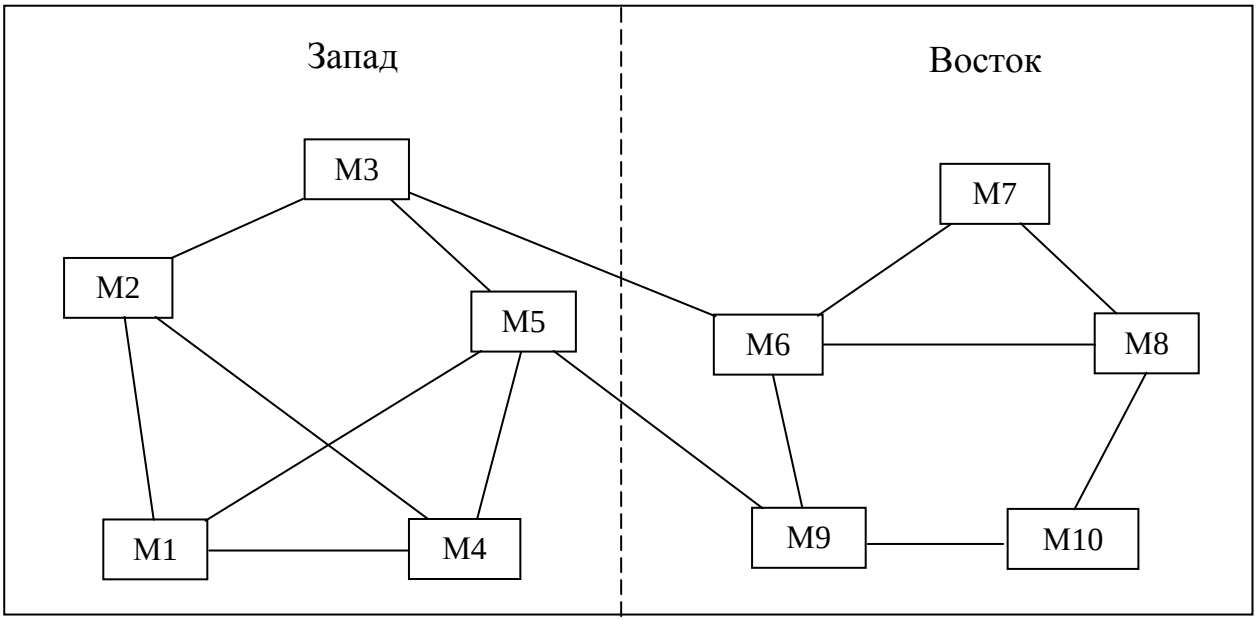


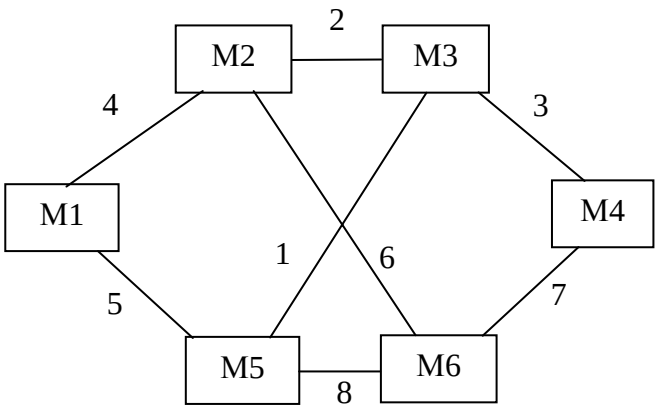
Рис. 4.9. Соединение частей подсети двумя линиями

Однако такое решение приведёт к неполному использованию наилучшего пути. Во избежание колебаний системы при выборе оптимального пути, по-видимому, лучше всего распределять нагрузку между несколькими линиями, пуская определённые части трафика по каждой из них.

4.14. Создание пакетов состояния линий

После формирования сообщения для передачи следующий шаг, выполняемый каждым маршрутизатором, заключается в создании пакета, содержащего все эти данные [3]. Пакет начинается с идентификатора отправителя, за которым следует порядковый номер, порядок и время создания (возраст), а также список соседей. Для каждого соседа указывается соответствующая ему задержка. Пример подсети приведен на рис. 4.10, *а*, на котором показаны задержки для каждой линии. Соответствующие пакеты состояния линий для всех шести маршрутизаторов показаны на рис. 4.10, *б*.

Процесс создания пакетов состояния линий несложен. Трудность заключается в выборе момента времени для их появления. Их можно генерировать периодически, через равные интервалы времени.



а

Пакеты состояния линий

M1	
№	Возраст
M2	4
M5	5

M2	
№	Возраст
M1	4
M3	2
M6	6

M3	
№	Возраст
M2	2
M4	3
M5	1

M4	
№	Возраст
M3	3
M6	7

M5	
№	Возраст
M1	5
M3	1
M4	8

M6	
№	Возраст
M2	6
M4	7
M5	8

б

Рис. 4.10. Пример создания пакетов состояния линий:
а – подсеть; *б* – пакеты состояния её линий

Другой вариант состоит в создании пакетов, когда происходит какое-либо значительное событие – например, линия или сосед выходит из строя или, наоборот, снова появляется в сети либо существенно изменяет свои свойства.

4.15. Распространение пакетов состояния линий

Самая сложная часть алгоритма заключается в распространении пакетов состояния линий [3]. По мере распространения и установки пакетов маршрутизаторы, получившие первые пакеты, начинают изменять свои маршруты. Соответственно, разные маршрутизаторы будут пользоваться разными версиями топологии, что может привести к противоречиям, появлению в маршрутах петель, недоступных машин, а также к другим проблемам.

Основная идея алгоритма распространения пакетов состояния линии состоит в использовании алгоритма заливки. Чтобы держать этот процесс под контролем, в каждый пакет помещают порядковый номер, увеличивающийся на единицу для каждого следующего пакета. Маршрутизаторы записывают в свою память все пары (источник, порядковый номер), которые им попадаются. Когда приходит новый пакет состояния линий, маршрутизатор ищет адрес его отправителя и порядковый номер пакета в своем списке. Если это новый пакет, он рассылается дальше по всем линиям, кроме той, по которой он пришел. Если же это дубликат, он удаляется. Если порядковый номер прибывшего пакета меньше, чем номер уже полученного пакета от того же отправителя, то такой пакет также удаляется как устаревший, поскольку очевидно, что у маршрутизатора есть более свежие данные.

С этим алгоритмом связано несколько проблем, о которых следует знать. Во-первых, если последовательный номер, достигнув максимально возможного значения, обнулится, возникнет путаница. Решение состоит в использовании 32-разрядных порядковых номеров. Даже если рассылать каждую секунду по пакету, то для переполнения 4-байтового целого числа понадобится 137 лет.

Во-вторых, если маршрутизатор выйдет из строя, будет потерян его порядковый номер. Если пакет будет снова загружен с нулевым порядковым номером, его пакеты будут игнорироваться как устаревшие.

В-третьих, может произойти искажение порядкового номера – например, вместо номера 4 будет принято число 65540 (ошибка в 1-м бите); в этом случае пакеты с 5-го по 65540-й будут игнорироваться некоторыми маршрутизаторами как устаревшие.

Решение этих проблем заключается в помещении в пакет после его порядкового номера возраста пакета и уменьшении его на единицу каждую секунду. Когда возраст уменьшается до нуля, информация от этого маршрутизатора удаляется. В нормальной ситуации новый пакет приходит, например, каждые 10 сек. Таким образом, сведения о маршрутизаторе устаревают, только когда он выключен (или в случае потери шести пакетов подряд, что маловероятно). Поле возраста также уменьшается на единицу каждым маршрутизатором во время начального процесса заливки, чтобы гарантировать, что ни один пакет не потеряется и не будет жить вечно.

Для повышения надёжности этого алгоритма используются некоторые усовершенствования. Когда пакет состояния линий приходит

на маршрутизатор для заливки, то не ставится сразу в очередь на отправку. Вместо этого он сохраняется в течение некоторого периода времени в области промежуточного хранения. Если за это время от того же отправителя успевает прийти ещё один пакет, маршрутизатор сравнивает их порядковые номера. Более старый пакет удаляется. Если номера одинаковые, то удаляется дубликат. Для защиты от ошибок на линиях связи между маршрутизаторами получение всех пакетов состояния линий подтверждается. Когда линия освобождается, маршрутизатор сканирует область промежуточного хранения, из которой выбираются для передачи пакеты или подтверждения.

Структура данных, используемая маршрутизатором М2 для работы с подсетью (см. рис. 4.10, *а*), показана на рис. 4.11. Каждый ряд здесь соответствует недавно полученному, но ещё не полностью обработанному пакету состояния линий. В таблице записываются адрес отправителя, порядковый номер, возраст и данные. Кроме того, в таблице содержатся флаги рассылки и подтверждений для каждой из трёх линий маршрутизатора М2 (к М1, М3 и М6 соответственно). Флаги отсылки означают, что этот пакет следует отослать по соответствующей линии. Флаги подтверждений означают, что нужно подтвердить получение этого пакета по данной линии.

Источник	№	Возраст	Флаги передачи			Флаги подтверждения			Данные
			М1	М3	М6	М1	М3	М6	
М1	21	60	0	1	1	1	0	0	
М6	21	60	1	1	0	0	0	1	
М5	21	59	0	1	0	1	0	1	
М3	20	60	1	0	1	0	1	0	
М4	21	59	1	0	0	0	1	1	

Рис. 4.11. Буфер пакетов маршрутизатора М2

Пакет состояния линий от маршрутизатора М1 пришёл напрямую, поэтому он должен быть отправлен маршрутизаторам М3 и М6, а подтверждение о его получении следует направить маршрутизатору М1, что и показывают флаговые биты. Аналогично, пакет от М6 следует переслать маршрутизаторам М1 и М3, а М6 отослать подтверждение.

Однако ситуация с третьим пакетом, полученным по сети от маршрутизатора М5, отличается. Пакет был получен дважды, по линиям М5-М1-М2 и М5-М6-М2. Следовательно, его нужно отослать только М3, но подтверждения выслать и М1, и М6, как указывают биты.

Если в то время, когда оригинал еще находится в буфере, прибывает дубликат пакета, значение битов должно быть изменено. Например, если копия состояния маршрутизатора М3 прибудет от М6 прежде, чем четвёртая строка таблицы будет разослана, шесть флаговых битов примут значение 100011, и это будет означать, что следует подтвердить получение пакета от М6, но не пересылать его М6.

4.16. Вычисление новых маршрутов

Собрав полный набор пакетов состояния линий, маршрутизатор может построить полный граф подсети, так как он располагает данными обо всех линиях [3]. На самом деле, каждая линия представлена даже дважды, по одному значению для каждого направления. Эти два разных значения могут усредняться или использоваться по отдельности.

Теперь для построения кратчайшего пути ко всем возможным адресатам может быть локально применён алгоритм Дейкстры. Результат вычислений может быть установлен в таблицах маршрутов, после чего можно возобновить нормальную работу маршрутизатора.

В подсети, состоящей из n маршрутизаторов, у каждого из которых k соседей, количество памяти, необходимой для хранения входной информации, пропорционально kn. Кроме того, может потребоваться много времени на обработку информации. В больших подсетях это может составлять проблему. Тем не менее, во многих практических ситуациях маршрутизация с учётом состояния линий работает вполне удовлетворительно.

Неисправности оборудования или программного обеспечения могут привести к очень серьёзным проблемам при использовании данного алгоритма. Например, если маршрутизатор заявит о существовании линии, которой у него в действительности нет, или наоборот, забудет о существовании имеющейся у него линии, граф подсети окажется неверным. Если маршрутизатор не сможет переслать пакеты или повредит их при пересылке, также возникнет проблема. Наконец, если у маршрутизатора закончится свободная память или он ошибётся в расчётах маршрутов, то возможны различные неприятности. При увеличении размера подсети до нескольких десятков или сотен тысяч маршрутизаторов вероятность выхода из строя одного из них перестаёт быть пренебрежимо малой. Всё, что можно здесь сделать – это попытаться ограничить вред, наносимый неизбежным выходом из строя оборудования.

Маршрутизация с учётом состояния линий широко применяется в современных сетях [3], поэтому следует сказать несколько слов о некоторых примерах протоколов, использующих данный алгоритм. Одним из таких протоколов является протокол OSPF, применяемый в Интернете.

Другим важным протоколом с учётом состояния линий является IS-IS (Intermediate System to Intermediate System – связь между промежуточными системами) – протокол, разработанный для сети DECnet и принятый впоследствии Международной организацией по стандартизации ISO для использования вместе с протоколом сетевого уровня CLNP, не требующим соединений. С тех пор он был модифицирован для поддержки также и других протоколов, в частности, IP. Протокол IS-IS используется в некоторых магистралях сети Интернет (включая старую магистраль NSFNET) и в цифровых сотовых системах, например, в CDPD. В сети Novell NetWare применяется разновидность протокола ISIS (NLSP) для маршрутизации IPX-пакетов.

В основе работы протокола IS-IS лежит распространение картины топологии маршрутизаторов, по которой рассчитываются кратчайшие пути. Каждый маршрутизатор в информации о состоянии линий сообщает доступные ему напрямую адреса сетевого уровня. Эти адреса могут быть адресами IP, IPX, AppleTalk или другими. Протокол IS-IS может даже осуществлять одновременную поддержку нескольких протоколов сетевого уровня.

Многие новшества, разработанные для протокола IS-IS, были приняты несколько лет спустя при разработке протокола OSPF. К ним относятся метод саморегуляции лавинного потока обновлений информации о состоянии линий связи, концепция выделенного маршрутизатора в локальной сети, а также метод вычисления и поддержки расщепления пути и умножения метрик. Соответственно, между протоколами IS-IS и OSPF почти нет разницы. Наиболее существенное различие между ними заключается в том, что способ кодирования в протоколе IS-IS, в отличие от OSPF, облегчает одновременную поддержку нескольких сетевых протоколов. Это свойство особенно важно в больших многопротокольных средах.

4.17. Иерархическая маршрутизация

Размер таблиц маршрутов, поддерживаемых маршрутизаторами, увеличивается пропорционально увеличению размеров сети. При этом

требуется не только большее количество памяти для хранения каждой таблицы, но и длительное время центрального процессора для её обработки. Кроме того, возрастает размер служебных пакетов, которыми обмениваются маршрутизаторы, что увеличивает нагрузку на линии. В определённый момент сеть может вырасти до таких размеров, при которых перестанет быть возможным хранение на маршрутизаторах записи обо всех остальных маршрутизаторах. Поэтому в больших сетях маршрутизация должна осуществляться иерархически, как это делается в телефонных сетях [3].

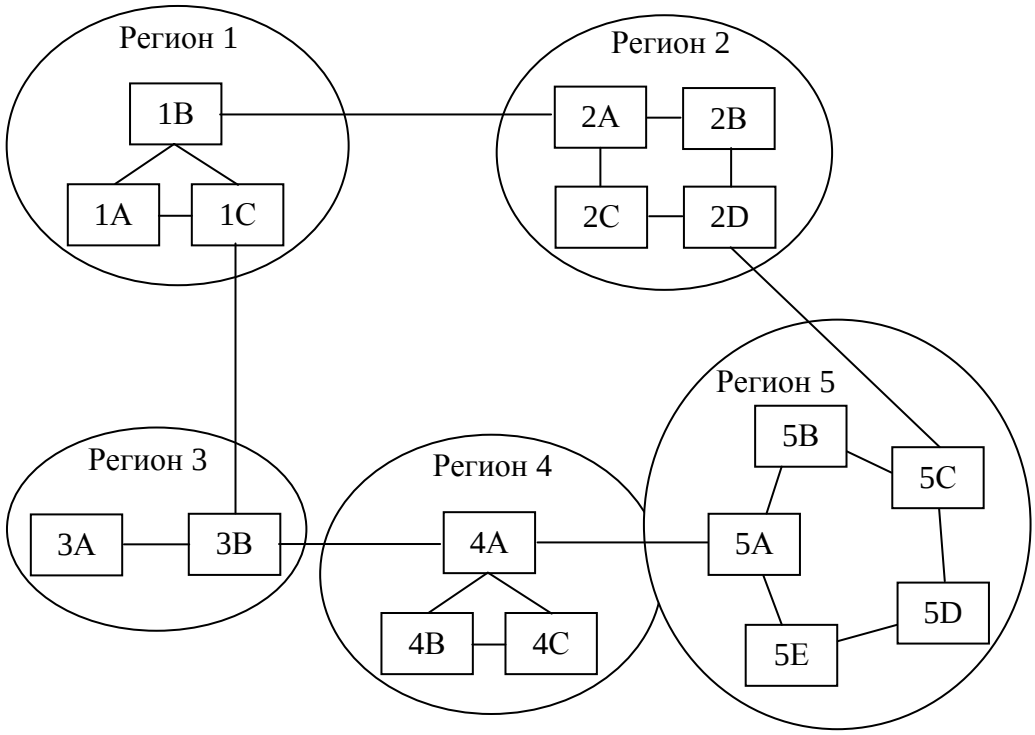
При использовании иерархической сетевой маршрутизации маршрутизаторы разбиваются на отдельные регионы. Каждый маршрутизатор знает все детали выбора маршрутов в пределах своей области, но ему ничего не известно о внутреннем строении других регионов, в том числе и смежных. При объединении нескольких сетей естественно рассматривать их как отдельные регионы, при этом маршрутизаторы одной сети освобождаются от необходимости знать топологию других сетей.

В очень больших сетях двухуровневой иерархии может оказаться недостаточно. Может потребоваться группировка регионов в кластеры, кластеры в зоны, зоны в группы, и т. д. На рис. 4.12 приведён количественный пример маршрутизации в двухуровневой иерархии с пятью регионами. Полная таблица маршрутизатора 1A состоит из 17 записей (рис. 4.12, *б*). При использовании иерархической маршрутизации (рис. 4.12, *в*) таблица, как и прежде, содержит сведения обо всех локальных маршрутизаторах, но записи обо всех остальных регионах концентрируются в пределах одного маршрутизатора, поэтому трафик во второй регион по-прежнему пойдёт по линии 1B - 2A, а во все остальные регионы – по линии 1C - 3B. При иерархической маршрутизации размер таблицы маршрутов уменьшается с 17 до 7 строк. Чем крупнее выбираются регионы, тем больше места в таблице экономится.

Выигрыш памяти весьма значителен, но за него приходится платить увеличением длины пути. Например, наилучший маршрут от 1A до 5C проходит через регион 2, но при использовании иерархической маршрутизации весь трафик в регион 5 направляется через регион 3, поскольку так лучше для большинства адресатов в регионе 5.

Когда единая сеть становится очень большой, возникает вопрос: сколько уровней должна иметь иерархия? Для примера рассмотрим подсеть с 720 маршрутизаторами. Если иерархии нет, то каждому

маршрутизатору необходимо поддерживать таблицу из 720 строк. Если подсеть разбить на 24 региона по 30 маршрутизаторов в каждом, тогда каждому маршрутизатору потребуется 30 локальных записей плюс 23 записи об удаленных регионах, итого – 53 записи.



а

Полная таблица для 1А

Назначение	Линия	Транзитные участки
1A	-	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

б

Иерархическая таблица для 1А

Назначение	Линия	Транзитные участки
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	3
4	1C	3
5	1C	4

в

Рис. 4.12. Иерархическая маршрутизация:
а – объединение региональных сетей в глобальную сеть; *б* – таблица маршрутизации для маршрутизатора 1А; *в* – иерархические таблицы для маршрутизации 1А

При выборе трёхуровневой иерархии, состоящей из 8 кластеров по 9 регионов из 10 маршрутизаторов, каждому маршрутизатору понадобится 10 строк в таблице для локальных маршрутизаторов, 8 строк для маршрутизации в другие регионы в пределах своего кластера, плюс 7 строк для удаленных кластеров, итого – 25 строк.

Ф. Камоун (F.Kamoun) и Л. Кляйнрок (L.Kleinrock) в 1979 г. показали, что оптимальное количество уровней иерархии для подсети, состоящей из N маршрутизаторов, равно ln N. При этом потребуется e ln Дозаписей для каждого маршрутизатора. Они также показали, что увеличение длины эффективного среднего пути, вызываемое иерархической маршрутизацией, довольно мало и обычно является приемлемым.

В ряде случаев источник сообщения посылает пакеты данных на множество адресатов или на всех адресатов сети (циркулярная передача). Эффективнее всего распространять соответствующие данные широковещательным способом, предоставляя возможность всем заинтересованным хостам получить их. Широковещанием называется рассылка пакетов по всем пунктам назначения.

Один из методов широковещательной маршрутизации не требует никаких особых способностей от подсети и используется просто для того, чтобы рассылать отдельные пакеты по всем направлениям. Он не только отнимает у подсети пропускную способность, но и требует, чтобы у источника пакета был полный список всех хостов. На практике это может быть единственной возможностью организации иерархии при передаче пакетов, но такой метод является наименее желательным.

Ещё одним очевидным кандидатом для иерархической рассылки пакетов является метод *заливки.* Хотя он плохо подходит для обычных двухточечных соединений, для широковещания это может быть серьёзный претендент, особенно если нет возможности применить один из методов, описываемых ниже. Проблема с применением заливки в качестве метода широковещания такая же, как с двухточечным алгоритмом маршрутизации: при заливке генерируется очень много пакетов и отнимается весьма существенная часть пропускной способности.

4.18. Широковещательная маршрутизация

Третий алгоритм называется многоадресной маршрутизацией [2, 3]. При использовании этого метода в каждом пакете содержится

либо список адресатов, либо битовая карта, показывающая предпочитаемые хосты назначения. Когда такой пакет прибывает на маршрутизатор, последний проверяет список, содержащийся в пакете, определяя набор выходных линий, которые потребуются для дальнейшей рассылки (линия может потребоваться в том случае, если она входит в оптимальный путь какого-либо из адресатов списка.) Маршрутизатором создаётся копия пакета для каждой из используемых исходящих линий. В неё включаются только те адресаты, для доступа к которым требуется данная линия. Таким образом, весь список рассылки распределяется между исходящими линиями. После определённого числа пересылок каждый из пакетов будет содержать только один адрес назначения и будет выглядеть как обычный пакет. Многоадресная маршрутизация в сети подобна индивидуально адресуемым пакетам, с той разницей, что в первом случае из нескольких пакетов, следующих по одному и тому же маршруту, только один платит полную стоимость, а остальные едут бесплатно.

Ещё один, четвёртый сетевой алгоритм широковещательной маршрутизации, в явном виде использует корневое дерево или любое другое связующее дерево. Связующее дерево представляет собой подмножество подсети, включающее в себя все маршрутизаторы, но не содержащее замкнутых путей. Если каждый маршрутизатор знает, какие из его линий принадлежат связующему дереву, он может отправить приходящий пакет по всем линиям связующего дерева, кроме той, по которой пакет прибыл. Такой метод оптимальным образом использует пропускную способность сети, порождая минимальное количество пакетов, требующихся для выполнения работы. Единственной проблемой этого метода является то, что каждому маршрутизатору необходимо обладать информацией о связующем дереве. Иногда такая информация доступна (например, в случае маршрутизации с учётом состояния линий), но иногда – нет (при маршрутизации по векторам расстояний).

Последний алгоритм широковещания представляет собой попытку приблизиться к поведению предыдущего алгоритма, даже когда маршрутизаторы ничего не знают о связующих деревьях. Лежащая в основе данного алгоритма идея, называющаяся продвижением по встречному пути, проста. Когда прибывает широковещательный пакет, маршрутизатор проверяет, используется ли та линия, по которой он прибыл, для нормальной передачи пакетов *источнику широковещания*. В случае положительного ответа велика вероятность того, что

широковещательный пакет прибыл по наилучшему маршруту и является, таким образом, первой копией, прибывшей на маршрутизатор. Тогда маршрутизатор рассылает этот пакет по всем линиям, кроме той, по которой он прибыл. Однако если пакет прибывает от того же источника по другой линии, он отвергается как вероятный дубликат.

Пример работы алгоритма продвижения по встречному пути показан на рис. 4.13. Слева изображена подсеть, посередине – входное дерево для маршрутизатора M9 этой подсети.

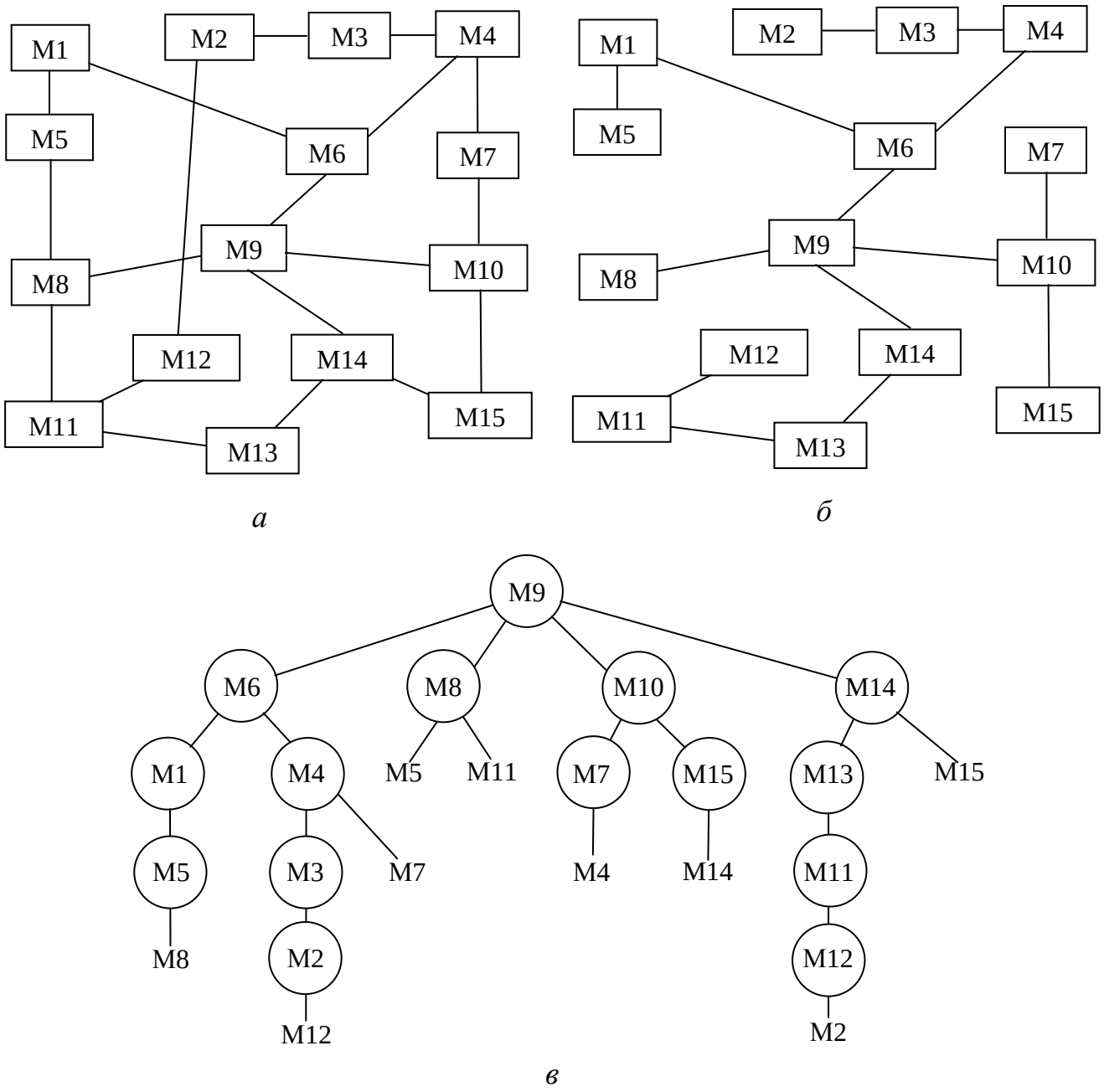


Рис. 4.13. Продвижение по встречному пути:
а – подсеть; *б* – связующее дерево; *в* – дерево, построенное методом продвижения по встречному пути

На первом транзитном участке маршрутизатор M9 посылает пакеты маршрутизаторам M6, M8, M10 и M14, являющимся вторым ярусом дерева. Все эти пакеты прибывают к M9 по предпочитаемым линиям (по пути, совпадающему с входным деревом), что обозначается кружками вокруг символов (рис. 4.13, *в*). На втором этапе пересылки формируются восемь пакетов – по два каждым маршрутизатором, получившим пакет после первой пересылки. Все восемь пакетов попадают к маршрутизаторам, не получавшим ранее пакетов, а пять из них приходят по предпочитаемым линиям. Из шести пакетов, формируемых на третьем транзитном участке, только три прибывают по предпочитаемым линиям (на маршрутизаторы M3, M12 и M11). Остальные оказываются дубликатами. После пяти транзитных участков широковещание заканчивается с общим количеством переданных пакетов равным 23, тогда как при использовании входного дерева потребовалось бы 4 транзитных участка и 14 пакетов.

Преимущество метода продвижения по встречному пути заключается в его эффективности при простоте реализации. Для использования этого метода маршрутизаторам не нужна никакая дополнительная информация о связующих деревьях. Не требуются и дополнительные расходы на список получателей или бит-карту в каждом распространяемом пакете, как в случае многоадресной рассылки. Также не требуется никакого специального механизма для прекращения процесса, как, например, в методе заливки (счётчик транзитных участков в каждом пакете и априорные сведения о составе сети или список уже встречавшихся пакетов от каждого источника).

4.19. Многоадресная рассылка

В ряде случаев, например, в ЛВС, одному процессу бывает необходимо послать сообщение всем остальным адресатам сети [3]. Если группа невелика, то можно просто послать каждому адресату отдельное сообщение. Если же группа достаточно большая, такая стратегия окажется весьма дорогостоящей. В ряде случаев может быть использовано широковещание, но применять его для информирования 1000 машин в сети, состоящей из миллиона узлов, неэффективно, поскольку большинство получателей будут не заинтересованы в данном сообщении (или наоборот, явно заинтересованы, но было бы крайне желательно от них эту информацию скрыть). Таким образом, требуется

способ рассылки сообщений строго определённым группам, довольно большим по численности, но небольшим по сравнению со всей сетью.

Передача сообщения членам такой группы называется многоадресной рассылкой, а алгоритм маршрутизации этой операции – многоадресной маршрутизацией. В этом пункте будет описан один из способов реализации многоадресной маршрутизации.

Многоадресной рассылке требуется управление группами, то есть способы создания и удаления групп, присоединения процесса к группе и ухода процесса из группы. Реализация данных задач, однако, не интересует алгоритм маршрутизации. Зато он заинтересован в том, чтобы процесс информировал свой хост о присоединении к какой-нибудь группе. Всегда важно, чтобы маршрутизаторы знали, какой хост к какой группе принадлежит. Для этого либо хост должен сообщать своим маршрутизаторам об изменениях в составе групп, либо маршрутизаторы должны сами периодически опрашивать свои хосты. В любом случае, маршрутизаторы узнают, какие из их хостов к каким группам принадлежат. Маршрутизаторы сообщают об этом своим соседям, и таким образом эта информация распространяется по всей подсети.

Для многоадресной рассылки каждый маршрутизатор сети рассчитывает связующее дерево, покрывающее все остальные маршрутизаторы подсети (рис. 4.14). В подсети с двумя группами маршрутизаторы соединены с хостами, принадлежащими к одной или обеим группам.

Когда процесс посылает группе многоадресный пакет, первый маршрутизатор изучает своё связующее дерево и отсекает у него линии, не ведущие к хостам, являющимся членами группы. В указанном примере (рис. 4.14, *в*, *г*) изображено усечённое связующее дерево для групп 1 и 2. Многоадресные пакеты рассылаются только вдоль соответствующего их группе усечённого связующего дерева. Разумеется, все остальные адресаты эти пакеты не получают, чем реализуется принцип разграничения доступа к информации.

Существует несколько способов усечения связующего дерева. Простейший способ оптимизации системы может применяться при использовании маршрутизации с учётом состояния линий, когда каждому маршрутизатору известна полная топология подсети, в том числе и состав групп. При этом из связующего дерева могут быть удалены маршрутизаторы, не принадлежащие к данной группе, начиная с конца каждого пути вплоть до корня дерева.

При маршрутизации по векторам расстояний может быть применена другая стратегия усечения дерева. Для многоадресной рассылки здесь применяется алгоритм продвижения по встречному пути.

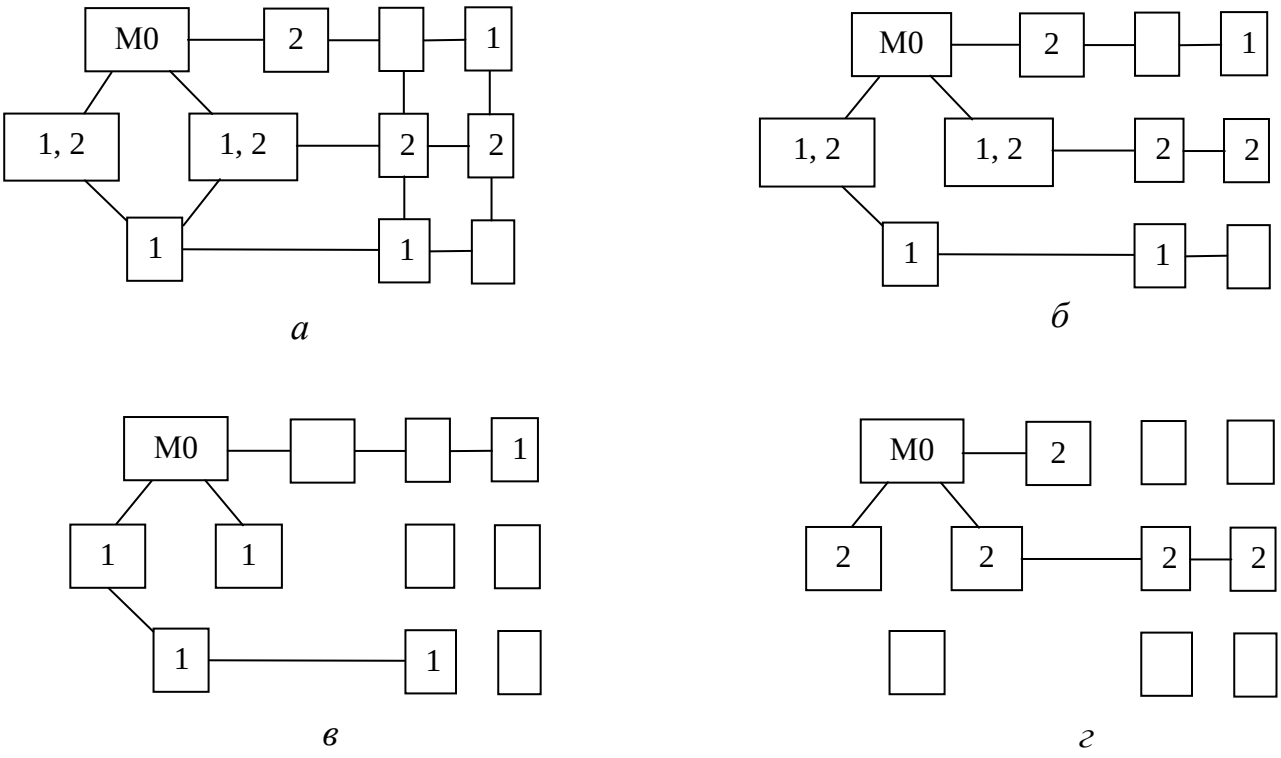


Рис. 4.14. Выборочная многоадресная рассылка:
а – подсеть; *б* – связующее дерево для маршрутизатора M0;
в – многоадресное дерево для группы 1; *г* – многоадресное дерево для группы 2

Когда многоадресное сообщение получает маршрутизатор, у которого нет хостов, входящих в группу, и линий связи с другими маршрутизаторами, он может ответить сообщением PRUNE (отсечь), информируя отправителя, что сообщения для данной группы ему больше посылать не нужно. Такой же ответ может дать маршрутизатор,
у которого нет хостов, входящих в группу, если он получил многоадресное сообщение по всем своим линиям. В результате подсеть постепенно рекурсивно усекается.

Недостаток данного алгоритма заключается в его плохой применимости к большим сетям. Предположим, что в сети есть n групп, каждая из которых в среднем состоит из m членов. Для каждой группы должно храниться m усеченных входных деревьев, то есть mn де-

ревьев для всей сети. При большом количестве групп для хранения всех деревьев потребуется много памяти.

Альтернативный метод использует деревья с основанием в сердцевине (Ballardie и др., 1993). В этом методе для каждой группы рассчитывается единое связующее дерево с корнем (ядром) около середины группы. Хост посылает многоадресное сообщение ядру группы, откуда оно уже рассылается по всему связующему дереву группы. Хотя это дерево не является оптимальным для всех источников, единое дерево для группы снижает затраты на хранение информации о нём в m раз.

4.20. Алгоритмы маршрутизации для мобильных хостов

На сегодняшний день миллионы людей обладают переносными компьютерами, и большинство из них желает читать свою электронную почту и получать доступ к нормальным файловым системам, находясь при этом в любой точке земного шара. Мобильные хосты привносят новое усложнение в и без того непростое дело выбора маршрутов в различных вычислительных сетях – чтобы направить пакет к мобильному хосту, его нужно сначала найти.

Фрагмент соединения региональных сетей и ЛВС с глобальной сетью показан на рис. 4.15 [3]. Глобальная сеть состоит из маршрутизаторов и хостов. С ней соединены локальные и региональные сети и беспроводные соты.

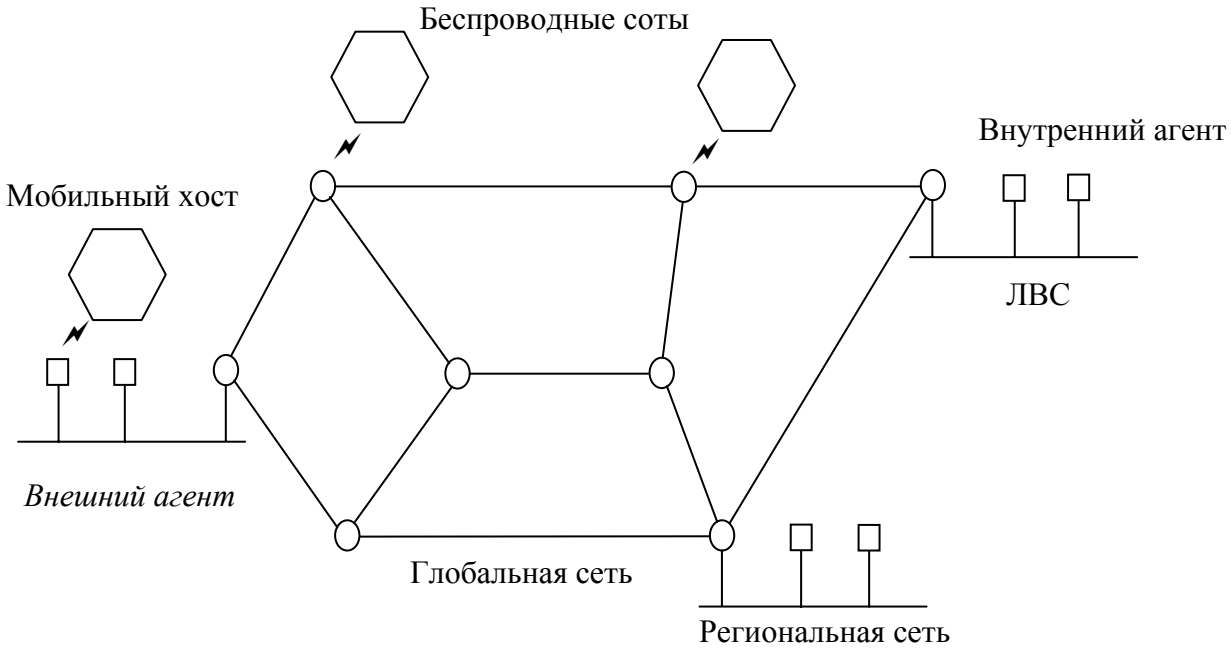


Рис. 4.15. Присоединение региональных сетей, ЛВС и беспроводных сот к глобальной сети

Хосты, которые никогда не перемещаются, называются стационарными. Они соединены с сетью поводами или волоконно-оптическими кабелями. Мигрирующие хосты являются в основном стационарными пользователями, но время от времени перемещаются с одного фиксированного места на другое и пользуются сетью только тогда, когда физически соединены с ней. Блуждающие хосты используют переносные компьютеры, и им требуется связь с сетью прямо во время перемещения в пространстве. Для обозначения этих двух категорий, то есть хостов, которые не имеют постоянного местоположения и тем не менее желают быть на связи, используется термин «мобильные хосты».

Предполагается, что у всех хостов есть постоянное домашнее местоположение, которое никогда не меняется. Кроме того, у хостов есть постоянный домашний адрес, которым можно воспользоваться для определения домашнего местоположения, аналогично тому, как телефонный номер (495) 234-55-35 обозначает г. Москву (код 495) и АТС района обслуживания (234). Целью маршрутизации в системах с мобильными хостами является обеспечение возможности передачи пакетов мобильным пользователям с помощью их домашних адресов. При этом пакеты должны эффективно достигать пользователей,

независимо от их расположения. Самое сложное здесь – найти пользователя.

В модели (см. рис. 4.15) система разделена на небольшие области, что означает локальную сеть или беспроводную соту. Каждая область может содержать одного или более внешних агентов, следящих за всеми мобильными пользователями, посещающими область. Кроме того, в каждой области имеется внутренний агент, следящий за временно покинувшими свою область пользователями.

Когда в области появляется новый или просто переместившийся в соту пользователь, – его компьютер должен зарегистрироваться в данной области (провести процедуру идентификации), связавшись с местным внешним агентом. Процедура регистрации обычно выглядит следующим образом:

1. Периодически каждый внешний агент рассылает пакет, объявляя таким образом о своём существовании и местонахождении (роль своего рода маяка). Вновь прибывший мобильный хост может ждать подобного сообщения, но может и сам, не дождавшись его, передать пакет с запросом о наличии внешнего агента в данной области.

2. Мобильный хост регистрируется в данной области, сообщая внешнему агенту свой домашний адрес, текущий адрес уровня передачи данных, а также информацию, подтверждающую его подлинность (аутентификация).

3. Внешний агент связывается с внутренним агентом мобильного пользователя и берёт его на обслуживание. Сообщение содержит адрес сети внешнего агента, а также информацию, подтверждающую подлинность мобильного хоста. Это позволяет убедить внутреннего агента в том, что мобильный хост действительно находится здесь.

4. Внутренний агент проверяет переданный ему идентификатор безопасности мобильного хоста, содержащий временной штамп, доказывающий, что идентификатор был создан буквально несколько секунд назад. Если аутентификация хоста проходит успешно, внутренний агент разрешает внешнему агенту продолжить связь.

5. Получив подтверждение от внутреннего агента, внешний агент заносит сведения о мобильном хосте в свою таблицу и сообщает ему, что он зарегистрирован.

В идеальном случае, покидая область, пользователь также должен сообщить об этом внешнему агенту, однако на практике многие пользователи, закончив работу, просто выключают свои компьютеры.

Когда пакет посылается мобильному пользователю, он направляется в его ЛВС на домашний адрес пользователя (рис. 4.16, этап 1). Отправитель направляет пакет хосту, который обычно находится по другую сторону распределённой сети. Пакеты, посланные в ЛВС мобильного хоста, перехватываются внутренним агентом, который узнаёт новое (временное) расположение мобильной станции и адрес внешнего агента локальной сети, в которой она в данный момент находится.

Затем внутренний агент выполняет два действия. Во-первых, он помещает пакет, предназначающийся мобильному пользователю, в поле данных внешнего пакета, который посылается внешнему агенту (этап 2). Такой приём называется туннелированием. Получив пакет, внешний агент извлекает из поля данных оригинальный пакет, который пересылает мобильному пользователю в виде кадра уровня передачи данных.

Далее внутренний агент сообщает пользователю-отправителю, что в дальнейшем следует не посылать пакеты мобильному хосту на домашний адрес, а вкладывать их в поле данных пакетов, явно адресованных внешнему агенту (этап 3). Последующие пакеты теперь могут направляться пользователю напрямую через внешнего агента (этап 4), полностью минуя домашний адрес мобильного пользователя.

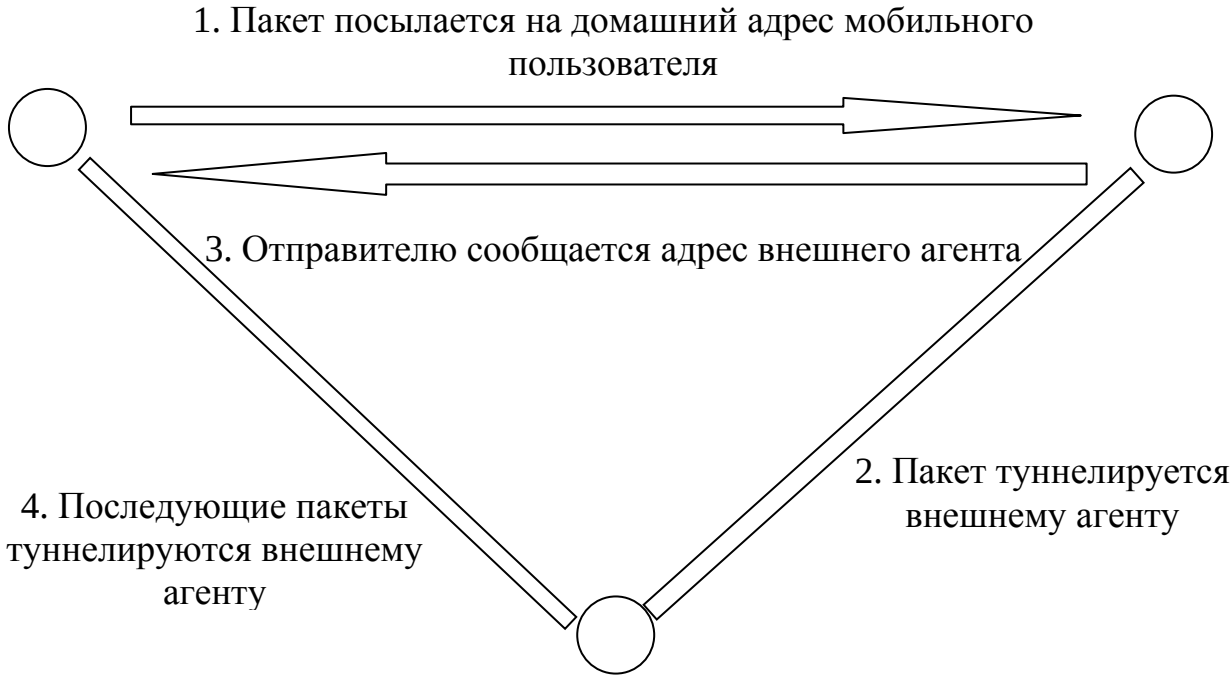


Рис. 4.16. Маршрутизация пакетов мобильным хостам

Предложенные схемы маршрутизации различаются в нескольких аспектах. Во-первых, в том, какая часть протокола выполняется маршрутизаторами, а какая – хостами, а также уровнем протоколов хостов. Во-вторых, в некоторых схемах маршрутизаторы записывают преобразованные адреса, поэтому они могут перехватывать и переадресовывать пакеты ещё до того, как те успевают дойти до домашнего адреса мобильного пользователя. В-третьих, в одних схемах каждому посетителю даётся уникальный временный адрес, а в других временный адрес ссылается на агента, обрабатывающего трафик для всех посетителей. В-четвёртых, схемы различаются способами переадресации пакетов. Один из способов заключается в изменении поля адреса получателя в пакете и передаче изменённого пакета. Есть системы, в которых весь пакет, включая домашний адрес, может быть помещён внутрь другого пакета, посылаемого по временному адресу. В любом случае, когда хост или маршрутизатор получает сообщение с указанием о пересылке на него всей информации, предназначаемой для хоста, у него могут возникнуть вопросы о достоверности и безопасности связи (аутентификация корреспондента).

4.21. Маршрутизация в специализированных сетях

Ранее уже было рассмотрено, как производится маршрутизация в случаях, когда станции мобильны, а маршрутизаторы стационарны. Более сложные ситуации возникают тогда, когда мобильны сами маршрутизаторы. Это возможно, например, в случаях наличия:

- военной техники на поле боя при отсутствии инфраструктуры;
- морских флотилий, находящихся в плавании;
- работников служб спасения в районах ЧС, не оборудованных в отношении связи или с разрушенными коммуникациями;
- собраний людей с портативными компьютерами при отсутствии в помещении сети.

Во всех подобных случаях каждый узел сети состоит из маршрутизатора и хоста одновременно, обычно они даже совмещены в пределах одного компьютера. Сети, состоящие из узлов, волею судеб оказавшихся недалеко друг от друга, называются мобильными специализированными сетями (MANET, Mobile Ad hoc networks) [3].

Основное отличие специализированных сетей от обычных проводных состоит в том, что все общепринятые законы, касающиеся фиксированной топологии, известных соседей, взаимосвязи между IP-адресом и расположением в специализированных сетях, перестают работать. Маршрутизаторы могут легко появляться в системе и так же легко из неё исчезать, появляясь в каком-то другом месте. В обычных сетях путь от маршрутизатора к какому-либо адресату продолжает оставаться реализуемым до тех пор, пока не произойдёт какой-нибудь сбой системы. В специализированных сетях топология постоянно меняется, а с ней меняется и предпочтительность (и даже реализуемость) путей. Причём это происходит спонтанно, безо всяких предупреждений. В таких условиях маршрутизация будет сильно отличаться от процесса в стационарных сетях.

Известно множество алгоритмов выбора маршрута для специализированных сетей. Один из наиболее интересных – это алгоритм AODV (Ad hoc On-demand Distance Vector – маршрутизация по требованию в специализированных сетях на основе вектора расстояний, Perkins and Royer, 1999).

AODV является разновидностью алгоритма Беллмана – Форда (Bellman – Ford, метод векторов расстояний), адаптированным для работы в мобильной среде и принимающим в расчет ограниченность пропускной способности и срока службы элементов питания – свойств, характерных для мобильных сетей. Ещё одной необычной характеристикой является то, что AODV – это алгоритм по требованию, то есть он вычисляет маршрут только в тот момент, когда появляется желающий отправить пакет тому или иному адресату.

4.22. Построение маршрута

Специализированная сеть в любой момент времени может быть описана с помощью графа узлов (маршрутизаторов и хостов) [4]. Два узла считаются соединенными (то есть между ними проведена дуга), если они могут связываться напрямую посредством радио. Поскольку у одного из них может быть более мощный передатчик, чем у другого, то возможна ситуация, когда узел *A* соединён с *B*, но *B* не соединён с *A*. Однако для простоты допустим, что все соединения симметричны. Следует заметить, что нахождение одного из узлов в зоне дей-

ствия другого ещё не означает наличия связи между ними. Их могут разделять холмы, здания и другие местные предметы, блокирующие соединение.

Пусть процессу, запущенному на узле *A*, необходимо отправить пакет на узел *I* (рис. 4.17). Алгоритм AODV на каждом узле ведёт таблицу, доступ к которой осуществляется с помощью поля адреса. Таблица содержит информацию об адресате, в том числе адрес ближайшего соседа, которому необходимо переслать пакет, чтобы он мог достичь пункта назначения. Допустим, *A* просматривает эту таблицу и не находит записи для *I*. Значит, нужно найти маршрут, ведущий к этому узлу. Итак, алгоритм начинает заниматься поисками маршрутов только тогда, когда они реально требуются. Это и делает его алгоритмом «по требованию».

Для поиска *I* узел *A* генерирует специальный пакет запроса маршрута ROUTE REQUEST и распространяет его по сети широковещательным способом. Сначала этот пакет достигает узлов *B* и *D* (рис. 4.17, *а*). Причиной установления именно узлами *B* и *D* соединения с *A* является то, что они могут получать пакеты от *A*. Например, *F* не соединён дугой с *A*, потому что он не может принимать радиосигнал от этого узла.

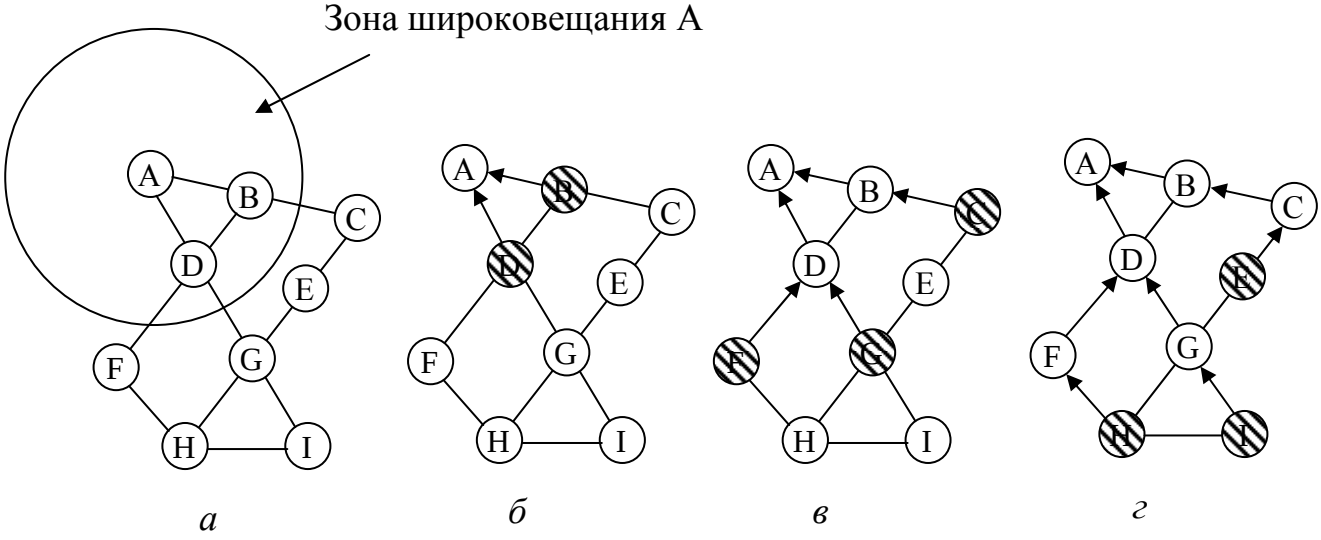


Рис. 4.17. Этапы пересылки пакета от узла A к узлу I:
а – зона широковещания A; *б* – состояние после получения узлами B и D широковещательного пакета от A; *в* – состояние после получения узлами C, F и G широковещательного пакета от A; *г* – состояние после получения узлами E, H и I широковещательного пакета от A. Штрихованные круги обозначают новых получателей

Формат пакета запроса маршрута показан на рис. 4.18. В пакете содержатся адреса источника и приёмника (обычно IP-адреса), с помощью которых можно понять, кто кого ищет. Также содержится поле *Идентификатор запроса,* которое представляет собой локальный счётчик, обновляемый каждым узлом независимо и инкрементирующийся всякий раз, когда распространяется пакет запроса маршрута. Поля *Адрес источника* и *Идентификатор запроса* вместе единственным образом идентифицируют пакет ROUTE REQUEST, что позволяет узлам обнаруживать и отвергать любые дубликаты.

Адрес отправителя	Идентификатор запроса	Адрес получателя	Порядковый номер отправителя	Порядковый номер получателя	Счётчик переходов

Рис. 4.18. Формат пакета ROUTE REQUEST

В дополнение к счётчику *Идентификатор запроса* каждый узел имеет второй счётчик, который инкрементируется всякий раз при отправке пакета для запроса маршрута или ответе на такой пакет. Его работа напоминает часы, он используется для того, чтобы можно было отличить новые маршруты от старых. Четвёртое поле – счётчик узла *A;* пятое – последнее значение порядкового номера пакета, полученного от *I* (оно равно 0, если такого пакета не было). Последнее поле – *Счётчик переходов* – запоминает количество пересылок, совершённых пакетом. В начале работы алгоритма оно равно нулю.

Когда пакет запроса маршрута прибывает на узел (например, на узлы *B* и *D*), с ним происходит следующее:

1. Пара значений полей *Адрес источника* и *Идентификатор запроса* ищется в таблице локальной истории. С их помощью можно выяснить, приходил ли уже этот запрос и обрабатывался ли он. Если обнаруживается, что пакет является дубликатом, он отвергается и его обработка прекращается. В противном случае, указанная пара значений заносится в таблицу истории, чтобы в будущем можно было обнаружить дубликаты. Обработка запроса продолжается.

2. Приёмник ищет адрес назначения в таблице маршрутов. Если известен достаточно свежий маршрут, отправителю посылается пакет наличия маршрута ROUTE REPLY, сообщающий ему о том, как можно достичь получателя быстрее. Свежий маршрут означает, что поле *Порядковый номер получателя* в таблице маршрутизации имеет зна-

чение большее или равное *Порядковому номеру получателя* из пакета запроса маршрута. Если оно меньше, значит, хранящийся в таблице маршрут является более старым, нежели предыдущий маршрут, имевшийся у отправителя к тому же пункту назначения. В этом случае выполняется пункт 3.

3. Поскольку у приёмника отсутствует свежий маршрут к адресату, он инкрементирует поле *Счётчик переходов* и вновь широковещательным образом распространяет по сети пакет запроса маршрута. Из пакета извлекаются данные и сохраняются в виде новой записи в таблице обратных маршрутов. Эти данные далее будут использоваться для построения обратного пути, по которому впоследствии необходимо будет послать ответный пакет отправителю. Стрелки (см. рис. 4.17) показывают возможные обратные маршруты. Для записи о только что созданном обратном пути запускается таймер. При наступлении тайм-аута запись удаляется.

Ни *B*, ни *D* не знают, где находится узел *I*, поэтому каждый из них создаёт обратный путь к *A*, как показано стрелками, и широковещательным способом распространяет пакет со *Счётчиком переходов*, установленным в единицу. Этот пакет от *B* достигает *C* и *D*. Узел *C* делает запись в таблице обратных путей и, в свою очередь, тоже широковещательным способом распространяет пакет далее. *A* отвергает пакет: для него это дубликат. Разумеется, и *B* отвергает пакеты, полученные от *D*. Тем не менее *F* и *G* принимают широковещательное сообщение от *D* и сохраняют его (см. рис. 4.17, *в*). После того, как *E, H* и *I* получают широковещательный пакет, запрос маршрута наконец достигает узла назначения (см. рис. 4.17, *г*). Несмотря на то, что распространение широковещательного пакета показано в виде трёх стадий, на самом деле, его рассылка разными узлами никак не координируется.

В ответ на пришедший запрос узел *I* генерирует пакет наличия маршрута ROUTE REPLY (рис. 4.19). Поля *Адрес отправителя*, *Адрес получателя* и *Счётчик переходов* копируются из ROUTE REQUEST, а *Порядковый номер получателя* берётся из собственного счётчика, хранящегося в памяти. Поле *Счётчик переходов* устанавливается в 0. Поле *Время существования* используется для управления реализуемостью маршрута пакета. Данный пакет распространяется методом одноадресной передачи на тот узел, с которого пришёл запрос маршрута. В данном случае, он уходит на узел *G*. Затем, в соответствии с установленным обратным путём, он попадёт на *D* и, наконец, на *A*.

При проходе каждого узла *Счётчик переходов* инкрементируется, так что узел-отправитель может увидеть, насколько далеко от него находится узел-получатель.

Адрес отправителя	Адрес получателя	Порядковый номер получателя	Счётчик переходов	Время существования

Рис. 4.19. Формат пакета ROUTE REPLY

Каждый узел, через который проходит пакет на обратном пути (к *A*), проверяет его. На основе информации пакета строится запись в локальной таблице маршрутов о пути к *I* при выполнении хотя бы одного из трёх условий:

1. Не известен ни один маршрут к *I*.

2. Последовательный номер для *I* в пакете ROUTE REPLY больше, чем значение в таблице маршрутизации.

3. Последовательные номера равны, но новый путь короче.

Таким образом, все узлы, стоящие на обратном пути к *A*, одновременно получают информацию о маршруте к узлу *I*. Это как бы побочный продукт построения маршрута пакетов для *A*. Узлы, получившие исходный пакет запроса маршрута, но не стоящие на обратном пути (узлы *B*, *C*, *E*, *F* и *H*), удаляют запись в таблице обратных маршрутов, когда ассоциированный с ней таймер достигает тайм-аута.

В больших вычислительных сетях алгоритмом генерируется много широковещательных пакетов даже для адресатов, расположенных довольно близко друг к другу. Число этих пакетов может быть уменьшено следующим образом. *Время жизни* IP-пакета устанавливается отправителем в значение, соответствующее ожидаемому диаметру сети, и декрементируется при каждой пересылке. Когда его значение становится равным 0, пакет отвергается, а не распространяется дальше.

При этом процесс поиска пути немного изменяется. Для обнаружения адресата отправитель рассылает пакет запроса маршрута с *Временем жизни*, равным 1. Если в течение разумного времени ответ не приходит, посылается ещё один запрос с *Временем жизни*, равным 2, и т. д. Таким образом поиск, начавшийся в какой-то локальной области, всё больше расширяет свой охват.

4.23. Обслуживание маршрута

Узлы могут перемещаться и выключаться, поэтому топология сети изменяется непредсказуемо. Например, если узел *G* выключится, *A* не поймёт, что путь к *I (ADGI)* больше не может быть реализован (см. рис. 4.17). В алгоритме должен быть предусмотрен механизм обеспечения живучести сети. Периодически все узлы рассылают сообщение приветствия *Hello*. Ожидается, что все узлы ответят на него. Если ответ не приходит, значит, сосед вышел из зоны действия и больше не связан с данным узлом. Аналогичным образом, если узел *A* пытается послать пакет соседу, который не отвечает, то он узнает, что связь с узлом *I* через узел *G* недоступна.

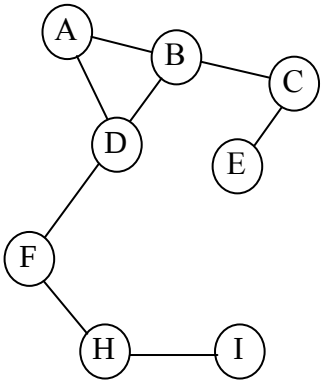
Эта информация используется для удаления нерабочих путей. Для каждого из возможных адресатов каждый узел сохранит историю о том, какие соседи снабжали узел пакетами для данных адресатов в течение последних секунд. Такие соседи называются активными соседями узла *N* для данного адресата. Узел *N* осуществляет сбор подобных сведений с помощью таблицы маршрутизации, которая, как известно, в качестве индекса использует адрес назначения. В этой таблице указан тот узел, на который нужно переслать пакет, чтобы он мог дойти до адресата. Кроме того, в ней имеются сведения об оставшемся числе переходов, последнем порядковом номере получателя, а также об активных соседях данного адресата. Вид возможной таблицы маршрутизации для узла *D* при топологии, рассмотренной выше, показан на рис. 4.20, *а*.

Когда какой-либо из соседей узла *N* становится недоступным, проверяется его таблица маршрутизации. Поскольку теперь нужно определить, к каким адресатам лежал путь через ушедший узел. Всем оставшимся активным соседям сообщается, что такие пути больше нельзя использовать и их следует удалить из таблиц маршрутизации. Активные соседи, в свою очередь, передают эти новости своим активным соседям, и так далее, пока все пути, зависевшие от ушедшего узла, не будут удалены из всех таблиц.

Адресат	Следующий переход	Расстояние	Активные соседи	Прочие поля
A	A	1	F, G	
B	B	1	F, G	
C	B	2	F	
E	G	2		
F	F	1	A, B	

G	G	1	A, B	
H	F	2	A, B	
I	G	2	A, B	

а



б

Рис. 4.20. Изменения топологии сети
при выходе из строя узла:
а – таблица маршрутизации для узла D перед выходом
из сети узла G; *б* – граф-схема сети после выхода из неё G

Исходя из рассмотренного выше примера, предположим, что *G* внезапно выключился (рис. 4.20, *б*). Когда *D* обнаруживает, что *G* ушёл из сети, он просматривает свою таблицу маршрутизации и видит, что *G* стоял на пути к *E, G* и *I*. Объединением активных соседей для данных адресатов является множество {*A*, *B*}. Другими словами, *A и B* содержат записи о маршрутах, проходящих через *G*, поэтому их нужно проинформировать о том, что эти маршруты больше не работают. *D* сообщает им об этом, посылая специальные пакеты, заставляющие их обновить свои таблицы соответствующим образом. Сам узел *D* удаляет записи для адресатов *E, G* и *I* из таблицы маршрутизации.

Из приведённого описания это, может быть, и не очевидно, но основная разница между AODV и алгоритмом Беллмана – Форда состоит именно в том, что узлы не занимаются периодической широковещательной рассылкой пакетов, содержащих полные таблицы маршрутизации. Благодаря этому более эффективно используется полоса пропускания и увеличивается время работы элементов питания.

4.24. Алгоритмы борьбы с перегрузкой

Когда количество пакетов, передаваемых одновременно по подсети (или её части), превышает некий пороговый уровень, производительность сети начинает снижаться. Такая ситуация называется перегрузкой (рис. 4.21).

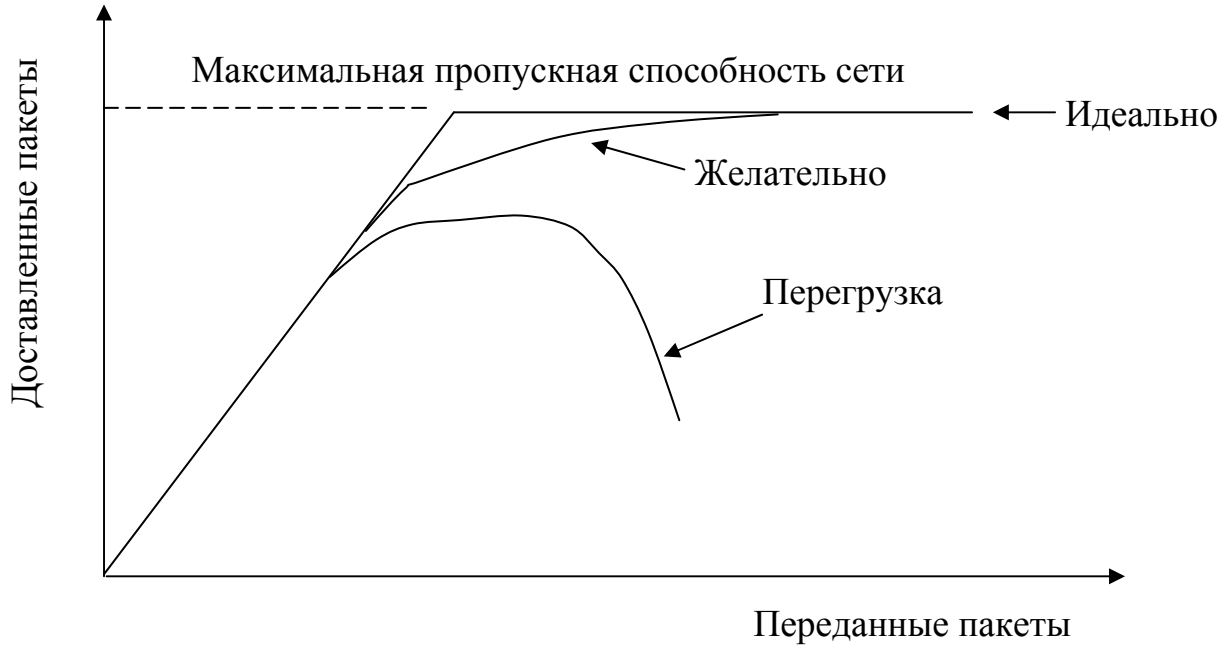


Рис. 4.21. Снижение производительности сети при перегрузке

Когда число пакетов, посылаемых хостами в сеть, не превышает её пропускной способности, все они доставляются адресатам (кроме небольшого процента повреждённых ошибками передачи). При этом количество доставленных пакетов пропорционально количеству посланных. Однако по мере роста трафика маршрутизаторы перестают успевать обрабатывать все пакеты и начинают их терять. При дальнейшем увеличении числа отправляемых пакетов ситуация продолжает ухудшаться. Когда число пакетов достигает максимального уровня, производительность сети начинает снижаться. При очень высоком уровне трафика производительность сети падает до совсем низкого уровня и практически никакие пакеты не доставляются.

Перегрузка может быть вызвана несколькими факторами. Если вдруг потоки пакетов начинают прибывать на маршрутизатор сразу по трём или четырём входным линиям и всем им нужна одна и та же выходная линия, то образуется очередь. Когда у маршрутизатора закончится свободная память для буферизации всех прибывающих па-

кетов, их негде будет сохранять и они начнут теряться. Увеличение объёма памяти маршрутизаторов может в какой-то степени помочь, но М. Нэгл (M. Nagle) в 1987 г. показал, что даже если у маршрутизаторов будет бесконечное количество памяти, ситуация с перегрузкой не улучшится, а, наоборот, ухудшится, так как к тому времени, когда пакеты доберутся до начала очереди, они уже запоздают настолько, что источником будут высланы их дубликаты. Все эти пакеты будут посланы следующему маршрутизатору, ещё более увеличивая нагрузку на всем протяжении маршрута к получателю.

Медленные процессоры также могут служить причиной заторов. Если центральные процессоры маршрутизаторов слишком медленно выполняют свои задачи, связанные с учётом, управлением очередями, обновлением таблиц, то очереди будут появляться даже при достаточно высокой пропускной способности линий. Аналогично, линии с низкой пропускной способностью также могут вызывать заторы в сети. Если заменить линии более совершенными, но оставить старые процессоры или наоборот, то такие действия обычно немного помогают, но часто просто приводят к сдвигу узкого места, вызванного несоответствием производительности разных частей системы. Проблема узкого места сохраняется до тех пор, пока компоненты системы не
будут должным образом сбалансированы.

Необходимо пояснить, в чём состоит разница между борьбой с перегрузкой и управлением потоком. Предотвращение перегрузки гарантирует, что подсеть справится с предлагаемым ей трафиком. Это глобальный вопрос, включающий поведение всех хостов и маршрутизаторов, процессов хранения и быстрой пересылки на маршрутизаторах, а также множество других факторов, снижающих пропускную способность подсети.

Управление потоком, напротив, относится к трафику между двумя конкретными станциями – отправителем и получателем. Задача управления потоком состоит в согласовании скорости передачи отправителя со скоростью, с которой получатель способен принимать поток пакетов. Управление потоком обычно реализуется при помощи обратной связи между получателем и отправителем.

Чтобы разница между этими двумя проблемами стала яснее, нужно представить себе оптоволоконную сеть с пропускной способно-

стью 1000 Гбит/с, по которой суперкомпьютер пытается передать персональному компьютеру файл со скоростью 1 Гбит/с. Хотя перегрузки сети в данной ситуации не наблюдается, алгоритм управления потоком довольно часто заставляет суперкомпьютер приостанавливать передачу, чтобы персональный компьютер мог успеть принять файл.

Другой пример. Рассмотрим сеть с промежуточным хранением, состоящую из 1000 больших компьютеров, соединённых линиями с пропускной способностью 1 Мбит/с. Одна половина компьютеров пытается передавать файлы другой половине со скоростью 100 кбит/с. Здесь проблема заключается уже не в том, что медленные получатели не успевают принимать данные, посылаемые им быстрыми отправителями, а просто в неспособности сети пропустить весь предлагаемый трафик.

Причина, по которой управление потоком и борьбу с перегрузкой часто путают, заключается в том, что алгоритмы борьбы с перегрузкой также используют обратную связь в виде специальных сообщений, посылаемых различным отправителям с просьбой передавать данные помедленнее, когда в сети появляются заторы. Таким образом, хост может получить просьбу замедлить передачу в двух случаях: когда с передаваемым потоком не справляется получатель или когда с ним не справляется вся сеть.

4.25. Общие принципы борьбы с перегрузкой

Многие проблемы, возникающие в сложных системах, таких, как компьютерные сети, следует рассматривать с точки зрения теории управления. При таком подходе все решения делятся на две группы [3]: без обратной связи и с обратной связью. Решения без обратной связи заключаются в попытках решить проблему с помощью улучшения дизайна системы, стремясь, таким образом, в первую очередь предотвратить возникновение самой ситуации перегрузки. Никаких корректирующих действий во время работы системы не предпринимается.

К методам управления без обратной связи относятся решения о том, когда конкретно разрешать новый трафик, когда отвергать пакеты и какие именно, а также составление расписаний для различных

участков сети. Общее в этих решениях то, что они не учитывают текущего состояния сети.

Решения с обратной связью, напротив, основываются на учёте текущего состояния системы. Этот подход состоит из трех следующих частей:

1. Наблюдение за системой с целью определения, где и когда произойдёт перегрузка.

2. Передача информации о перегрузке в те места, где могут быть предприняты соответствующие действия.

3. Принятие необходимых мер при работе системы для устранения перегрузки.

При наблюдении за состоянием подсети с целью обнаружения перегрузки могут измеряться различные параметры. Среди них следует выделить следующие: процент пакетов, отвергаемых из-за отсутствия свободного места в буфере, средняя длина очереди, процент пакетов, переданных повторно по причине истекшего времени ожидания подтверждения, среднее время задержки пакетов и среднеквадратичное отклонение задержки пакетов. Во всех случаях увеличивающиеся значения параметров являются сигналами о растущей перегрузке.

Второй этап борьбы с перегрузкой состоит в передаче информации о перегрузке от места её обнаружения туда, где могут быть приняты какие-то меры по её устранению. Очевидное решение заключается в том, чтобы маршрутизатор, обнаруживший перегрузку, пересылал источнику или источникам трафика пакет с извещением о наличии проблемы. Такие пакеты, конечно, окажут дополнительную нагрузку на сеть как раз в тот момент, когда нагрузку необходимо снизить.

Существуют, однако, и другие решения. Например, можно зарезервировать в каждом пакете бит или поле, которые будут заполняться маршрутизаторами при достижении перегрузкой порогового уровня. Таким образом, соседи этого маршрутизатора будут предупреждены о том, что на данном участке сети наблюдается перегрузка.

Ещё один метод состоит в том, что хосты или маршрутизаторы периодически посылают пробные пакеты, явно спрашивая друг друга о перегрузке. Собранная таким образом информация может затем использоваться для выбора маршрутов в обход тех участков сети, где возникла проблема с перегрузкой. Так, в больших городах радиостанции сообщают слушателям о заторах на дорогах в надежде, что води-

тели выберут маршруты для своих пакетов (то есть машин) в обход пробок.

Все системы с обратной связью предполагают, что получившие информацию о перегрузке в сети хосты и маршрутизаторы предпримут какие-либо действия для устранения перегрузки. Чтобы данная схема работала, необходимо тщательно настроить временные параметры. Если каждый раз, когда два пакета приходят одновременно, какой-нибудь маршрутизатор будет останавливать движение, а простояв без работы 20 мкс, он же будет внезапно открываться, система войдёт в состояние постоянных незатухающих колебаний. С другой стороны, если маршрутизатор не изменит своего состояния и для большей надёжности станет ждать 30 мин., прежде чем выдать оповещение в сеть, то механизм борьбы с перегрузкой станет реагировать слишком медленно, чтобы приносить вообще какую-либо пользу. Для правильной работы необходимо некоторое усреднение, однако правильный выбор значения постоянной времени является нетривиальной задачей.

Известны различные алгоритмы борьбы с перегрузкой. Д. Янг (D. Yang) и А.Н. Редди (A.N. Reddy, 1995) разработали специальный метод классификации этих алгоритмов. Они начали с того, что разделили все методы на алгоритмы с обратной связью и без неё, как уже описывалось ранее. Затем они разделили алгоритмы без обратной связи на работающие у отправителя и у получателя. Алгоритмы с обратной связью также были разделены на две подкатегории: с явной и неявной обратной связью. В алгоритмах с явной обратной связью от точки возникновения перегрузки в сети в обратном направлении посылаются пакеты, предупреждающие о заторе. В алгоритмах с неявной обратной связью источник приходит к выводу о наличии перегрузки, основываясь на локальных наблюдениях, – например, по значению интервала времени, требующегося процессу для получения подтверждения.

Наличие перегрузки означает, что нагрузка временно превысила возможности ресурсов данной части системы. Есть два решения данной проблемы: увеличить ресурсы системы или снизить нагрузку. Например, подсеть может использовать телефонные линии с модемами, чтобы увеличить пропускную способность между определёнными точками. В спутниковых системах большую пропускную способность часто даёт увеличение мощности передатчика. Распределение трафика по нескольким маршрутам вместо постоянного использования одного

и того же, пусть даже оптимального пути, также может позволить ликвидировать местную перегрузку. Наконец, для увеличения пропускной способности сети в случае серьёзных заторов могут быть задействованы запасные маршрутизаторы, которые обычно применяются для повышения устойчивости системы в случае сбоя.

Однако иногда увеличить пропускную способность бывает невозможно либо она уже увеличена до предела. В таком случае единственный способ борьбы с перегрузкой состоит в уменьшении нагрузки. Для этого существует несколько способов, включая отказ в обслуживании или снижение уровня обслуживания некоторых или всех пользователей, а также составление более четкого расписания потребностей пользователей в обслуживании.

4.26. Стратегии предотвращения перегрузки

Системы без обратной связи разработаны в первую очередь для предотвращения перегрузки, а не для борьбы с уже имеющей место перегрузкой. Они пытаются достичь своей цели, используя соответствующие стратегии на разных уровнях (табл. 4.2) [3].

Начнём рассмотрение различных стратегий с канального уровня. Стратегия повторной передачи определяет, насколько быстро у отправителя истекает время ожидания подтверждения и что он передаёт после того, как время ожидания истекло. Приоритетный отправитель, у которого время ожидания срочных сообщений истекает слишком быстро и который повторно посылает все неподтверждённые пакеты с помощью алгоритма возврата на *n*, окажет более сильную нагрузку на сеть, чем второстепенный отправитель, использующий выборочный повтор. Тесно связана с этим стратегия хеширования. Если получатели просто игнорируют все пакеты, приходящие не в том порядке, то все проигнорированные пакеты придётся передавать позднее ещё раз, что окажет дополнительную нагрузку на сеть.

Таблица 4.2

Стратегии предотвращения перегрузки

Уровень	Стратегии
Транспортный	Повторная передача. Хеширование пакетов, приходящих с нарушением порядка. Подтверждения. Управление потоком. Определение тайм-аутов
Сетевой	Виртуальные каналы против дейтаграмм в составе подсети. Очереди пакетов и обслуживания. Игнорирование пакетов. Алгоритм маршрутизации. Управление временем жизни пакетов
Канальный	Повторная передача. Хеширование пакетов, приходящих с нарушением порядка. Подтверждения. Управление потоком

Стратегия подтверждений также влияет на перегрузку. Если каждый пакет немедленно подтверждается получателем, то пакеты с подтверждениями (квитанции) образуют дополнительный трафик. Однако если подтверждения добираются обратно с попутным потоком кадров, то количество трафика в сети снижается, но увеличивается среднее время получения подтверждений, что может, в свою очередь, вызвать увеличение повторно переданных пакетов вследствие истечения времени ожидания подтверждений. Более жёсткая схема управления потоком (например, с небольшим размером окна) уменьшает скорость передачи данных и помогает бороться с перегрузкой.

Существует также зависимость перегрузки от того, является ли сетевой уровень дейтаграммным или он основан на виртуальных каналах, так как многие алгоритмы борьбы с перегрузкой работают только в подсетях с виртуальным каналами. Политика очередей пакетов и обслуживания определяет количество очередей у каждого маршрутизатора – например, может существовать одна общая очередь для всех линий, или по очереди для каждой линии, или какой-нибудь комбинированный вариант. Она определяет также порядок обработки пакетов (например, поочередно или в порядке приоритетов). Политика игнорирования пакетов является правилом, определяющим набор пакетов, которые не будут обрабатываться, если не хватает памяти. Хорошо продуманная стратегия может облегчить симптомы пере-

грузки, тогда как неудачная политика может даже ухудшить ситуацию.

Хороший алгоритм выбора маршрута может помочь избежать локальной перегрузки, перераспределяя трафик по всем линиям, тогда как неудачный алгоритм может направить слишком большое количество пакетов по одной линии и вызвать затор. Наконец, управление временем жизни пакетов определяет, как долго пакет может перемещаться по сети, прежде чем он будет проигнорирован очередным маршрутизатором. Если это время слишком велико, то потерянные пакеты могут забивать сеть, однако если время жизни пакета слишком мало, то пакеты не будут успевать достичь адресата, что приведёт к необходимости повторных передач.

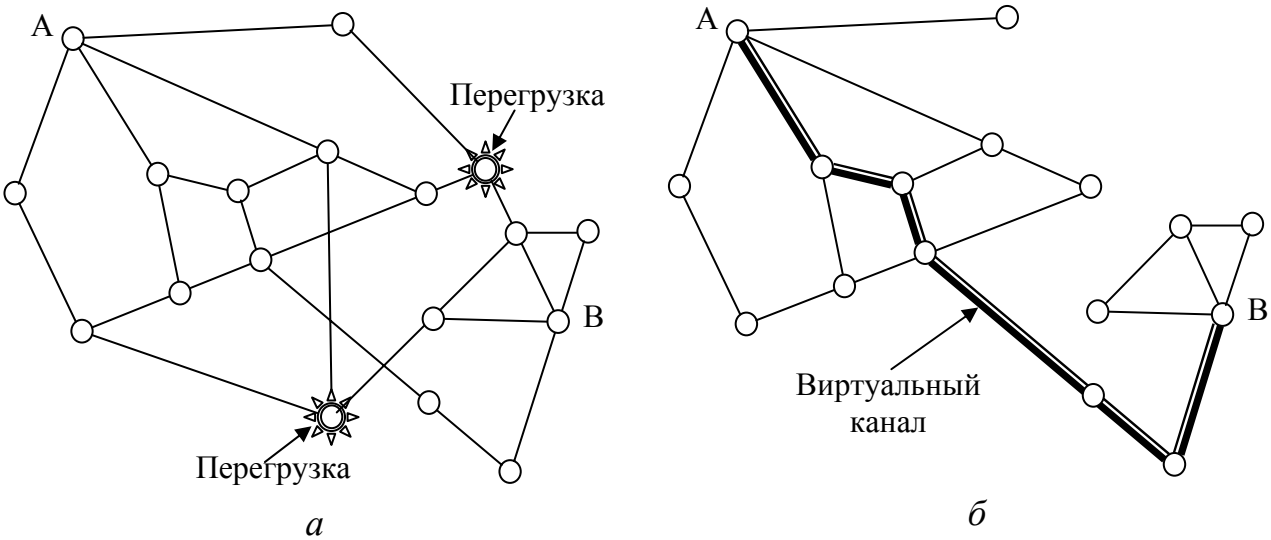
На транспортном уровне применяются те же стратегии, что и на канальном, но к ним добавляется проблема определения времени ожидания подтверждения, что на транспортном уровне осуществить значительно сложнее, поскольку время пересечения всей сети предсказать значительно сложнее, чем время передачи пакета от какого-либо маршрутизатора до его соседа. Если этот интервал слишком короток, будут высылаться излишние повторные пакеты, а если слишком велик – перегрузка снизится, но увеличится задержка в случае потери пакета.

4.27. Борьба с перегрузкой в подсетях виртуальных каналов

Широко применяемым методом недопущения ухудшения уже начавшейся перегрузки является управление допуском. Идея этого метода проста: когда приходит сигнал о перегрузке, никакие новые виртуальные каналы не создаются до тех пор, пока проблема не разрешится [3]. То есть любые попытки установить новые соединения транспортного уровня пресекаются. Понятно, что если пустить в сеть, в которой уже возникла перегрузка, дополнительных пользователей, то ситуация только ухудшится. Метод предполагает временный отказ в доступе новых пользователей к сети. Это вызывает неудобство из-за снижения оперативности системы передачи в целом. Абонент по неизвестным причинам должен ждать обслуживания. Но, с точки зрения функциональности сети, метод дешёв, надёжен и практичен. В обычных телефонных системах при перегрузке коммутатора абонент поднимает трубку и не слышит сигнала готовности станции.

Альтернативный подход заключается в том, что создание новых виртуальных каналов разрешается, но эти каналы тщательно прокладываются в обход заторов. Для примера рассмотрим подсеть (рис. 4.22), в которой два маршрутизатора перегружены.

Предположим, что хост, соединённый с маршрутизатором *A*, предполагает установить соединение с хостом, соединённым с маршрутизатором *B*. В нормальных условиях это соединение прошло бы через один из перегруженных маршрутизаторов. Чтобы этого избежать, подсеть уменьшается (рис. 4.22, *б*). При этом из неё удаляются перегруженные маршрутизаторы и все их линии связи. Толстой линией показан возможный маршрут виртуального канала в обход перегруженных маршрутизаторов.



а *б*

Рис. 4.22. Перегрузка подсети и её устранение:
а – перегруженная подсеть; *б* – устранение перегрузки организацией виртуального канала

Другое решение, связанное с виртуальными каналами, состоит в достижении соглашения между хостом и подсетью во время установки виртуального канала. Этот протокол обычно описывает объём и форму трафика, требуемое качество обслуживания и другие параметры. В качестве выполнения своей части соглашения подсеть обычно резервирует ресурсы на пути следования создаваемого канала. К этим ресурсам относятся память для буферов и таблиц маршрутизаторов и пропускная способность линий. При таком подходе возникновение перегрузки в новом виртуальном канале маловероятно, так как все необходимые ресурсы были зарезервированы и их доступность гарантируется.

Подобное резервирование может выполняться как в виде постоянной стандартной процедуры, так и в виде специального действия, выполняемого только при возникновении перегрузки. Недостаток постоянного резервирования заключается в том, что на эту процедуру расходуются вычислительные ресурсы. Если шесть виртуальных каналов, которым разрешено использовать по 1 Мбит/с, проходят по одной и той же физической линии с пропускной способностью 6 Мбит/с, линия должна быть помечена как полная, хотя маловероятно, что все шесть виртуальных каналов передают данные одновременно, да ещё и используют всю доступную пропускную способность. Следовательно, платой за резервирование будет неиспользованная пропускная способность (низкий КПД сети).

4.28. Сброс нагрузки

Сбросом нагрузки сети называется простое игнорирование маршрутизаторами пакетов, которые они не могут обработать [3]. Своим происхождением в специальной литературе этот термин обязан системам электроснабжения, где он означает отключение в случае перегрузок отдельных участков во избежание выхода из строя всей системы. Обычно такое происходит в морозные зимние дни, когда потребности в электроэнергии для обогревателей резко возрастают.

Маршрутизатор, заваленный пакетами, может выбирать пакеты просто случайным образом, но обычно имеются более оптимальные варианты. Выбор пакета, который будет отвергнут, может зависеть от приложения, пересылающего этот пакет. Для передачи файла более старый пакет ценится выше нового, так как отвержение пакета номер 6 и сохранение пакетов с номерами с 7-го по 10-й может привести к тому, что получатель запросит ещё раз пакеты с 6-го по 10-й (если получатель просто отвергает все пакеты, приходящие не в том порядке). В файле, состоящем из 12 пакетов, выбрасывание 6-го пакета может потребовать повторной передачи пакетов с 7-го по 12-й, тогда как выбрасывание пакета номер 10 может потребовать повторной передачи только пакетов с 10-го по 12-й. Но для мультимедийных приложений, напротив, новый пакет важнее старого.

Чтобы сделать этот алгоритм ещё разумнее, необходимо участие в нём отправителей. Во многих приложениях одни пакеты могут быть значительно важнее других. Например, некоторые алгоритмы сжатия

видеосигнала периодически посылают полный кадр, а последующие кадры представляют собой лишь карты изменений относительно последнего полного кадра. В таком случае потеря пакета, содержащего разностный сигнал, не так страшна, как потеря полного кадра. Точно так же при передаче страницы, содержащей текст и рисунок, потеря линии пикселов рисунка может остаться почти незамеченной, тогда как потеря строки текста крайне нежелательна.

Для реализации интеллектуальной стратегии выбрасывания части информации приложения должны помечать свои пакеты классами приоритетов, соответствующими их важности. В этом случае маршрутизаторы смогут сначала выбросить пакеты нижнего класса, затем следующего за ним и т. д. Конечно, при отсутствии стимула все будут помечать свои пакеты не иначе как ОЧЕНЬ ВАЖНО, ПОТЕРЯ НЕДОПУСТИМА.

Стимулом может служить стоимость обслуживания, то есть пересылка пакетов низкоприоритетным классом может быть дешевле, чем высокоприоритетным. В качестве альтернативы источникам сообщений может быть ультимативно предложено отправлять высокоприоритетные пакеты только в условиях низкого трафика, а с повышением загрузки сети прекращать их отправку.

Ещё один вариант состоит в разрешении хостам превышать пределы, указанные в соглашении, заключённом при создании виртуального канала (например, использовать большую пропускную способность, чем договаривались), но при условии, что весь дополнительный трафик будет помечаться как низкоприоритетный. Такая стратегия весьма удачна, поскольку более эффективно использует свободные ресурсы, разрешая хостам пользоваться ими, пока это никому не мешает, но не закрепляя за ними этого права.

4.29. Борьба с флуктуациями

Для таких приложений, как аудио- и видеопередача, не так уж важно, 20 или 30 мс занимает доставка пакетов, – до тех пор, пока время доставки будет постоянно. Колебание (среднеквадратичное отклонение) времени доставки пакетов называется флуктуацией. Если одни пакеты будут доставляться за 20, а другие – за 30 мс, изображение или звук начнут дрожать. В этом случае говорят о наличии сильных флуктуаций. Однако внутри системы может существовать правило, что 99 % пакетов должны быть доставлены с задержкой в диа-

пазоне от 24,5 до 25,5 мс, и качество при этом будет вполне приемлемым.

Выбранный диапазон должен быть, конечно, выполнимым. При вычислении времени задержки необходимо принимать во внимание время передачи по каналу со скоростью света, минимальную задержку при прохождении маршрутизаторов, а также некоторые другие неизбежные задержки.

Для ограничения флуктуации должно быть как можно точнее вычислено ожидаемое время пересылки по каждому транзитному участку пути [3]. Получив пакет, маршрутизатор проверяет, насколько пакет опаздывает или опережает график. Эта информация хранится в каждом пакете и обновляется каждым маршрутизатором. Если пакет приходит с опережением графика, он удерживается в течение требуемого интервала времени. Если же пакет запаздывает, маршрутизатор пытается отправить его дальше как можно быстрее.

Алгоритм, определяющий, какие из пакетов отправить первыми по выходной линии, всегда может выбрать пакет, больше всего отстающий от расписания. При этом пакеты, опережающие график, замедляются, а опаздывающие пропускаются в первую очередь, что в обоих случаях уменьшает флуктуации времени доставки пакетов.

В некоторых приложениях, таких, как видео по требованию, флуктуации могут быть снижены путём сохранения пакетов в буфере приёмника с их последующей выдачей для отображения. При этом сеть не обязана работать в реальном масштабе времени. Тем не менее в приложениях, которые должны обеспечивать межпользовательское взаимодействие в реальном масштабе времени (например, в Интернет-телефонии или видеоконференциях), задержка, связанная с буферизацией, совершенно недопустима.

4.30. Требования к качеству обслуживания

Рассмотренные ранее методы направлены на уменьшение перегрузок и повышение производительности в сетях передачи данных. Однако с ростом доли мультимедийной информации таких специализированных параметров оказывается явно недостаточно. Необходимо обеспечить гарантированное качество обслуживания сетей и улучшения протоколов.

Последовательность пакетов, передающихся от источника к приемнику, называется потоком. При этом в сетях, ориентированных на

соединение, все пакеты потока следуют по одному и тому же маршруту, а в сетях без установления соединения они могут идти разными путями. Каждому потоку требуются определённые условия, которые можно охарактеризовать следующими четырьмя основными параметрами: надёжность, задержка, флуктуация и пропускная способность. Все вместе они формируют то, что называется качеством обслуживания (QoS – Quality of Service), необходимым потоку (табл. 4.3) [3].

Таблица 4.3

Требования некоторых приложений к качеству обслуживания

Приложение	Надёжность	Задержка	Флуктуации	Пропускная способность
Электронная почта	Высокая	Низкая	Слабые	Низкая
Передача файлов	Высокая	Низкая	Слабые	Средняя
WEB-доступ	Высокая	Средняя	Слабые	Средняя
Удалённый доступ	Высокая	Средняя	Средние	Низкая
Аудио по заказу	Низкая	Низкая	Сильные	Средняя
Видео по заказу	Низкая	Низкая	Сильные	Высокая
Телефония	Низкая	Высокая	Сильные	Низкая
Видеоконференция	Низкая	Высокая	Сильные	Высокая

Первые четыре приложения предъявляют высокие требования к надёжности. Некорректная доставка битов должна быть исключена. Обычно это достигается подсчётом контрольной суммы для каждого пакета и её проверкой у получателя. Если пакет во время передачи был испорчен, подтверждение о его доставке не высылается, и источник вынужден передавать его повторно. Такая стратегия обеспечивает высокую надёжность. Четыре последних (аудио/видео) приложения весьма толерантны к ошибкам, поэтому здесь нет никаких вычислений и проверок контрольных сумм.

Приложения, занимающиеся передачей файлов, включая электронную почту и видео, не чувствительны к задержкам. Даже если все пакеты будут доставляться с задержкой в несколько секунд, ничего страшного не произойдёт. Однако интерактивные приложения, например, обеспечивающие WEB-доступ или удалённый доступ, – к задержкам более критичны. Что касается приложений, работающих в реальном масштабе времени, их требования к задержкам очень строги. Если при телефонном разговоре все слова собеседников будут приходить с задержкой ровно 2 с, пользователи сочтут такую связь неприемлемой. С другой стороны, проигрывание видео- или аудиофайлов, хранящихся на сервере, допускает наличие некоторой задержки.

Первые три приложения спокойно отнесутся к неравномерной задержке доставки пакетов, а при организации удалённого доступа этот фактор имеет более важное значение, поскольку при сильных флуктуациях символы на экране будут появляться скачками. Видео- и особенно аудиоданные чрезвычайно чувствительны к флуктуациям. Если пользователь просматривает видео, доставляемое на его компьютер по сети, и все кадры приходят с задержкой 2 с, всё нормально. Однако если время передачи колеблется от одной до двух секунд, то результат будет неприемлем. При прослушивании звукозаписей будут заметны флуктуации даже в несколько миллисекунд.

Наконец, приложения могут иметь различные потребности в пропускной способности. При передаче электронной почты или при удалённом доступе высокая пропускная способность не требуется, а вот для передачи видеоданных любых типов необходима высокая производительность сети.

В сетях ATM принята следующая классификация потоков по требованиям к качеству обслуживания:

1. Постоянная битовая скорость (например, телефония).
2. Переменная битовая скорость в реальном времени (например, сжатые видеоданные при проведении видеоконференций).
3. Переменная битовая скорость не в реальном времени (например, просмотр фильмов через Интернет).
4. Доступная битовая скорость (например, передача файлов).

Такое разбиение по категориям может оказаться полезным и для других целей, и для других сетей. Постоянная битовая скорость – это попытка моделирования проводной сети путём предоставления фиксированной пропускной способности и фиксированной задержки. Битовая скорость может быть переменной, например, при передаче сжатого видео, когда одни кадры удаётся сжать в большей степени, чем другие. Кадр, содержащий множество разноцветных деталей, сожмётся, скорее всего, плохо, и на его передачу придётся потратить много битов, тогда как кадр, запечатлевший белую стену, сожмется очень хорошо. Приложениям типа электронной почты нужно принципиальное наличие хоть какой-нибудь битовой скорости, они не чувствительны к задержкам и флуктуациям, поэтому говорят, что этим приложениям требуется доступная битовая скорость.

4.31. Избыточное обеспечение и буферизация

Проще всего обеспечить такую ёмкость маршрутизаторов сети, буферной памяти и такую пропускную способность, при которых пакеты без затруднений перемещались бы по каналам и узлам [3]. Проблема здесь одна: такое решение обходится очень дорого. Со временем разработчики начинают понимать, какие параметры являются необходимыми и достаточными, и тогда такой подход оправдывает себя. Можно сказать, что телефонная сеть является системой с избыточным обеспечением. Довольно редко бывает, чтобы вы подняли трубку и не услышали гудка. Дело в том, что в систему заложена настолько большая пропускная способность, что превысить её оказывается тяжело.

Потоки можно сохранять в буферной памяти на принимающей стороне, перед тем, как доставлять потребителю. Буферизация не сказывается на надёжности и пропускной способности, но сказывается на увеличении задержки. Зато с её помощью можно снизить уровень флуктуации. При передаче аудио и видео по требованию именно флуктуация представляет собой основную проблему, и буферизация помогает решить её.

При значительной флуктуации поток пакетов, доставляемый адресату, записывается в память буфера. Пакеты прибывают с различным временем задержки, причём это время нарастает по мере поступления пакетов. Прибывающие пакеты буферизируются на клиентской машине.

В установленное время начинается воспроизведение, при этом пакеты с 1-го по n-й уже находятся в буфере, поэтому их можно оттуда извлекать через равные интервалы и воспроизводить. Если какой-либо пакет задержался настолько, что его невозможно воспроизвести вовремя, то выдача пакетов приостанавливается до его прибытия. Возникает задержка в воспроизведении информации (например, музыки или фильма). Проблему можно решить увеличением задержки начала выдачи пакетов, но для этого потребуется буфер большей ёмкости. Все коммерческие WEB-сайты, на которых содержится потоковое видео или аудио, используют проигрыватели, которые начинают воспроизведение только после примерно десятисекундной буферизации.

4.32. Формирование трафика

В приведённом выше примере источник выдаёт пакеты через фиксированные интервалы времени, однако бывает, что этот процесс не является столь равномерным. Это может приводить к перегрузке сети. Неравномерный выходной поток – это обычное дело для серверов, поддерживающих множество потоков и различные виды действий, таких, как быстрая прокрутка вперёд и назад, а также идентификация пользователей сети. Подход, описанный ранее (буферизация), не всегда можно применить (например, при видеоконференциях). Тем не менее, если бы удалось заставить серверы (и хосты в целом) передавать данные с предсказуемой скоростью, качество обслуживания было бы выше. Рассмотрим метод формирования трафика, сглаживающий выходной трафик на стороне сервера, а не на стороне клиента [3].

При формировании трафика происходит регулирование средней и пиковой *скоростей* передачи данных. Изучавшиеся нами ранее протоколы скользящего окна ограничивают количество данных, посылаемых сразу, но не скорость, с которой они посылаются. Когда устанавливается виртуальный канал, пользователь и подсеть (то есть клиент и оператор связи) договариваются об определённой схеме (то есть форме) трафика для данного канала. Иногда это действие называется соглашением об уровне обслуживания. До тех пор, пока клиент выполняет свою часть условий соглашения и посылает пакеты не чаще оговоренного в договоре графика, оператор связи обязуется доставлять их в определённый срок. Формирование трафика снижает перегрузку и, таким образом, помогает оператору связи выполнять свои обязательства. Подобные договоренности не столь важны при передаче файлов, но весьма существенны при передаче данных в режиме реального времени, как, например, для аудио- и видеосвязи, которые плохо переносят перегрузку.

В результате, при использовании метода формирования трафика клиент сообщает оператору связи: «Мой трафик передачи будет выглядеть следующим образом. Сможете ли вы это обеспечить?». Если оператор соглашается, то возникает вопрос о том, как он будет сообщать клиенту, что тот соблюдает соглашение, и что делать, если клиент нарушит договор. Наблюдение за потоком трафика называется политикой трафика. Договориться о форме трафика и регулировать его впоследствии легче в подсетях с виртуальными каналами, чем

в дейтаграммных подсетях. Тем не менее, даже в дейтаграммных подсетях можно применить те же идеи к соединениям транспортного уровня.

4.33. Алгоритм «дырявого ведра»

Представьте себе ведро с маленькой дырочкой в днище. Независимо от скорости, с которой вода наливается в ведро, выходной поток обладает постоянной скоростью, когда в ведре есть вода, и нулевой скоростью, когда ведро пустое. Кроме того, когда ведро наполняется, вся лишняя вода выливается через край и теряется (то есть не попадает в выходной поток под дырочкой).

Та же самая идея применима к пакетам [3]. Принцип таков: каждый хост соединяется с сетью через интерфейс, содержащий «дырявое ведро», то есть конечную внутреннюю очередь (рис. 4.23).

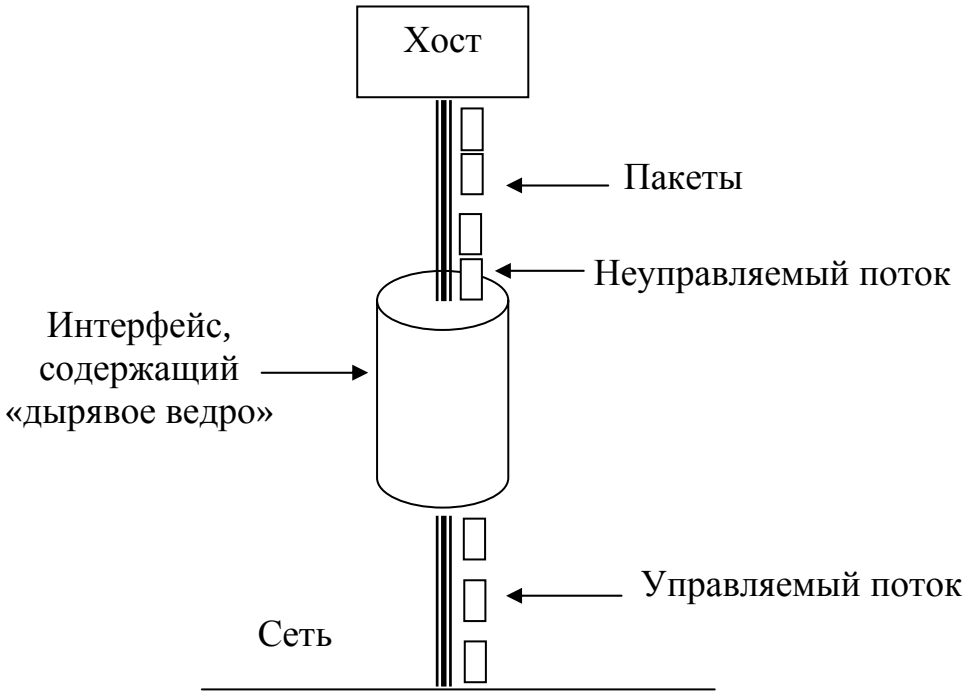


Рис. 4.23. **«**Дырявое ведро» с пакетами

Если пакет появляется в очереди, когда та полная, то пакет игнорируется. Другими словами, если несколько процессов хоста пытаются послать пакеты, когда в очереди уже стоит максимально допустимое число пакетов, новый пакет игнорируется. Такой интерфейс может быть реализован как аппаратно, так и программно-операционной

системой хоста. Он был предложен Е.С. Тернером (J.S. Turner, 1986) и назван алгоритмом «дырявого ведра». По сути, это не что иное, как однолинейная система массового обслуживания с постоянным временем обслуживания

Хосту разрешается посылать в сеть один пакет за один такт. Опять же, это может быть реализовано интерфейсной картой либо операционной системой. Этот механизм преобразует неравномерный поток пакетов от процессов пользователя в равномерный поток пакетов в сети, сглаживая пики и значительно снижая вероятность перегрузки.

Когда размер всех пакетов одинаков (например, в ячейках ATM), этот алгоритм может применяться, как описано ранее. Однако при использовании пакетов переменного размера часто бывает лучше ограничивать количество байтов, переданных в сеть за такт, чем передавать один пакет за такт. Так, если правилом установлена передача 1024 байтов за тактовый интервал, то за этот период могут быть переданы в сеть либо один пакет размером 1024 байта, либо два пакета по 512 байтов, либо четыре пакета по 256 байтов и т. д. Если оставшееся количество байтов меньше размера следующего пакета, следующий пакет должен ждать начала следующего такта.

Реализация исходного алгоритма «дырявого ведра» проста. «Ведро» состоит из конечной очереди. Когда прибывает пакет и в очереди есть место, пакет добавляется к очереди, в противном случае пакет игнорируется. Если очередь не пуста, то в течение каждого тактового интервала в сеть передается по одному пакету.

Алгоритм «дырявого ведра» со счётчиком байтов реализуется почти также. В каждом тактовом интервале значение счётчика устанавливается равным *n*. Если размер первого пакета в очереди меньше текущего значения счётчика, он передаётся, а значение счётчика уменьшается на его размер. Если значение счётчика ещё достаточно велико, могут быть посланы и другие пакеты. Когда значение счётчика становится меньше размера следующего пакета в очереди, передача прекращается до следующего такта, после чего всё начинается сначала, а остаток счётчика обнуляется.

В качестве примера можно представить, что компьютер производит данные со скоростью 25 млн. байт в секунду (200 Мбит/с) и что сеть также работает на этой скорости. Однако маршрутизаторы могут поддерживать эту скорость передачи данных лишь на коротких интервалах (пока не заполнится их буферная память). В течение больших интервалов времени они могут обеспечить не более 2 млн. байт

в секунду. Теперь предположим, что данные поступают пачками по 1 млн. байт, одна пачка продолжительностью 40 мс в каждую секунду. Чтобы уменьшить среднюю скорость до 2 Мбайт/с, можно воспользоваться алгоритмом «дырявого ведра» с выходной скоростью p = 2 Мбайт/с и ёмкостью C = 1 Мбайт. Это означает, что пачки до 1 Мбайт могут обрабатываться без потерь и что такие пачки будут передаваться в сеть за 500 мс, независимо от того, как быстро они приходят.

4.34. Алгоритм «маркерного ведра»

Алгоритм «дырявого ведра» формирует строгий выходной поток с постоянной скоростью, не зависящей от неравномерности входного потока. Для многих приложений было бы лучше при поступлении больших пакетов данных немного увеличивать выходную скорость. Требуется более гибкий алгоритм, не теряющий данные. Одним из таких алгоритмов является алгоритм «маркерного ведра». В этом алгоритме «ведро» содержит маркеры, создаваемые через равные интервалы времени ΔT секунд (рис. 4.24) [3].

Здесь изображено «ведро» с тремя маркерами и пятью пакетами, стоящими в очереди. Чтобы передать один пакет, требуется удалить один маркер. На рис. 4.24, *б* видно, что три из пяти пакетов прошли в сеть, а оставшиеся два пакета остались ждать двух новых маркеров.

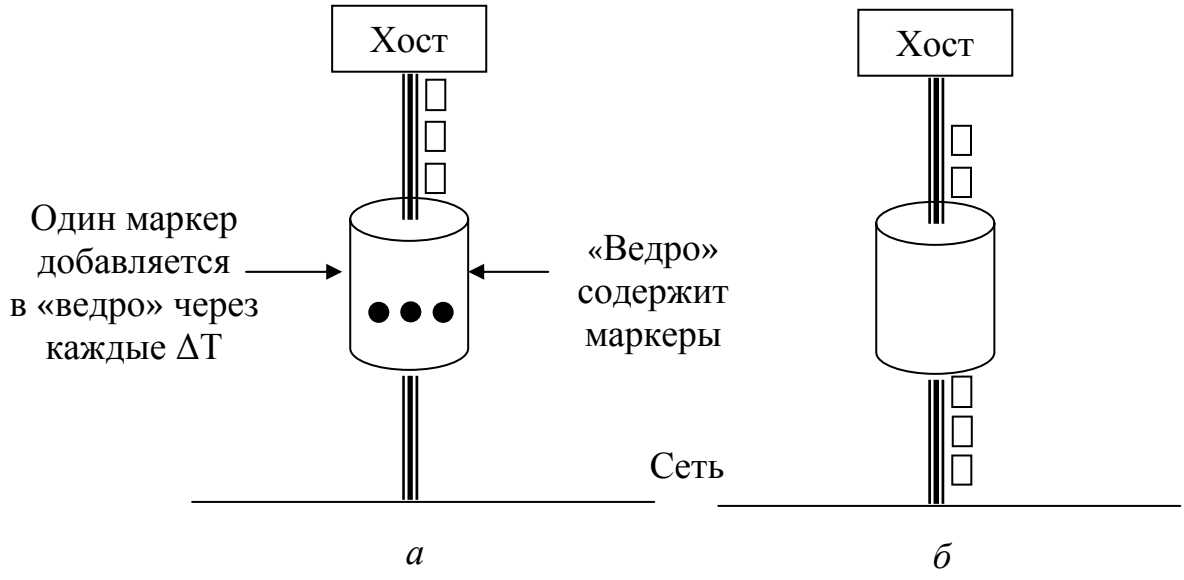


Рис. 4.24. Алгоритм «маркерного ведра»:
а – до прохождения пакетов; *б* – после прохождения пакетов

Алгоритм «маркерного ведра» формирует трафик не так, как алгоритм «дырявого ведра». Алгоритм «дырявого ведра» не позволяет простаивающим хостам запасаться впрок разрешениями на передачу больших пакетов. Алгоритм «маркерного ведра» разрешает запасаться маркерами до определённого размера «ведра» *n*. Это свойство означает, что пачки (пакеты) с величиной до *n* могут быть переданы в сеть сразу, что создаёт некоторую неравномерность в выходном потоке, но обеспечивает быструю реакцию на неожиданные входные пачки.

Ещё одно различие двух алгоритмов состоит в том, что при переполнении «маркерного ведра» алгоритм игнорирует маркеры, но никогда не отвергает пакеты. Алгоритм «дырявого ведра», напротив, при переполнении выбрасывает сами пакеты.

Возможен вариант алгоритма, при котором маркер может предоставлять право пересылать не один пакет, а *k* байтов. Пакет пересылается только при наличии достаточного числа маркеров, чтобы покрыть его длину. Лишние маркеры сохраняются для будущего использования.

Сетевые алгоритмы «дырявого» и «маркерного ведра» могут использоваться не только для регулирования выхода хостов, но и для сглаживания трафика между маршрутизаторами. Эти два алгоритма различаются тем, что применение алгоритма «маркерного ведра» может заставить маршрутизатор остановить передачу пакетов, что приведёт к потере данных.

Реализация исходного сетевого алгоритма «маркерного ведра» подразумевает наличие переменной, считающей маркеры. Счётчик увеличивается на единицу через равные интервалы времени ΔT и уменьшается при посылке пакета. Когда счётчик уменьшается до нуля, передача пакетов останавливается. В варианте с учётом количества переданных байтов счётчик увеличивается на *k* байтов каждые ΔT секунд и уменьшается на размер переданного пакета.

Суть алгоритма «маркерного ведра» в том, что он допускает передачу данных пачками, но ограничивает длительность пачки. Например, «маркерное ведро» имеет ёмкость 250 кбайт. Маркеры появляются с частотой, соответствующей выходной скорости 2 Мбайт/с. Предположим, что «маркерное ведро» заполнено, когда прибывает пакет данных размером 1 Мбайт. «Ведро» может быть освобождено с максимальной скоростью 25 Мбайт/с примерно за 11 мс. Затем оно должно уменьшить скорость передачи до 2 Мбайт/с, пока не будет передан весь входной пакет данных.

При расчёте длительности выходной пачки (на максимальной скорости) нужно учитывать, что пока «ведро» опорожняется, появляются новые маркеры. При длительности пачки 5 секунд, ёмкости «маркерного ведра» C байтов, скорости появления маркеров p байт/с и максимальной выходной скорости M байт/с, очевидно, что максимальное количество переданных байтов в пачке будет равно $C + pS$ байтов. В то же время количество байтов, переданных в пачке с максимальной скоростью, равно MS. Таким образом, $C + pS = MS$.

Решив это уравнение, получим: $5 = \frac{C}{M - p}$. При указанных выше параметрах C = 250 кбайт, M = 25 Мбайт/с и p = 2 Мбайт/с получаем длительность пачки около 11 мс.

Недостатком алгоритма «маркерного ведра» является слишком большая скорость передачи данных при опустошении «ведра», несмотря на то, что длительность пачки можно регулировать тщательным подбором p и M. Часто бывает желательно уменьшить пиковую скорость, не возвращаясь при этом к скорости алгоритма «дырявого ведра».

Один из способов получения более гладкого трафика состоит в помещении «дырявого ведра» после «маркерного ведра». Скорость «дырявого ведра» должна быть выше минимальной скорости «маркерного ведра» p, но ниже максимальной скорости сети. Управление такими схемами может оказаться непростым. Сеть должна имитировать алгоритм и гарантировать, что пакетов и байтов посылается не больше, чем разрешено. Тем не менее эти методы позволяют формировать сетевой трафик, приводя его к более управляемому виду и обеспечивая тем самым выполнение требований к качеству обслуживания.

4.35. Резервирование ресурсов

Возможность управления трафиком – это неплохой начальный шаг в деле обеспечения гарантированного качества обслуживания. Однако на самом деле использование этих методов неявно означает, что все пакеты в потоке должны следовать по одному и тому же пути. При распределении их случайным образом между несколькими сетевыми маршрутизаторами невозможно что-либо гарантировать. Следовательно, между источником и приёмником должно быть установлено

нечто вроде виртуального канала, и все пакеты, принадлежащие данному потоку, должны следовать по указанному маршруту [3].

Если есть особый путь, по которому направляется поток, становится возможным резервирование ресурсов вдоль этого пути, что позволяет гарантировать доступность необходимой ёмкости. Резервироваться могут три типа ресурсов:

1. Пропускная способность.
2. Буферное пространство.
3. Время центрального процессора.

Наиболее очевидно резервирование пропускной способности. Если потоку необходима скорость 1 Мбит/с, а исходящая линия может работать со скоростью 2 Мбит/с, то направить три потока с такими параметрами по этой линии не удастся. То есть резервирование пропускной способности означает предотвращение предоставления канала большему числу абонентов, чем канал может обработать.

Вторым дефицитным ресурсом является буферное пространство. Когда прибывает пакет, он обычно фиксируется на сетевой интерфейсной карте (это действие управляется аппаратно). Затем программному обеспечению сетевого маршрутизатора необходимо скопировать пакет в буфер оперативной памяти и поставить содержимое этого буфера в очередь на отправку по выбранной исходящей линии. Если буферное пространство недоступно, входящий пакет приходится игнорировать, поскольку его просто негде сохранить. Для обеспечения хорошего качества обслуживания можно резервировать некоторую часть буферной памяти под конкретный поток, чтобы ему не пришлось бороться за буфер с другими потоками. Тогда при передаче потока ему всегда будет предоставляться выделенная часть буфера, вплоть до некоторого максимума.

Наконец, время центрального процесса – это ещё один очень ценный ресурс. На что расходуется время работы процессора в маршрутизаторе? На обработку пакетов. Поэтому существует предельная скорость, с которой маршрутизатор может обрабатывать пакеты. Необходимо быть уверенным в том, что процессор не перегружен.

На первый взгляд, кажется, что если на обработку пакета уходит, к примеру, 1 мкс, то маршрутизатор способен управиться с миллионом пакетов за секунду. Однако это предположение ошибочно, так как при доставке потока всегда есть промежутки времени, в течение которых информация не передаётся. Если центральному процессору

для совершения своей работы важен каждый отдельный такт, то пропуск нескольких тактов из-за молчания на линии приведёт к накоплению невыполненных заказов, от которых невозможно избавиться.

Однако даже если нагрузка несколько меньше теоретической ёмкости, всё равно могут образовываться очереди и возникать задержки. Рассмотрим ситуацию, когда пакеты прибывают нерегулярно, со средней скоростью прибытия λ пакетов в секунду. Время, необходимое процессору на обработку каждого пакета, также меняется, но в среднем составляет μ пакетов в секунду. Предположим, что как скорость прибытия, так и скорость обслуживания имеют пуассоновское распределение. Тогда, используя теорию массового обслуживания, можно доказать, что средняя задержка *T*, присущая пакету, составляет

$$T = \frac{1}{\mu} \cdot \frac{1}{1 - \lambda / \mu} = \frac{1}{\mu} \cdot \frac{1}{1 - \rho},$$

где $\rho = \lambda/\mu$ – коэффициент использования центрального процессора. Первый сомножитель $1/\mu$ – это задержка при отсутствии конкуренции. Второй сомножитель представляет собой дополнительную задержку, возникающую в результате конкурентной борьбы с другими потоками. Например, если λ = 950 000 пакетов/с, а μ = 1 000 000 пакетов/с, тогда ρ = 0,95, и средняя задержка каждого пакета составляет 20 мкс вместо 1 мкс. Эти подсчёты учитывают и задержку доставки, и задержку обработки: при малом трафике отношение $\lambda/\mu \rightarrow 0$. Если на пути потока стоят, к примеру, 30 маршрутизаторов, то одна только задержка обслуживания составит 600 мкс.

4.36. Управление доступом

После применения рассмотренных алгоритмов входящий трафик будет в виде хорошо сформированного и, возможно, следующего по единому маршруту потока. На пути потока можно заранее резервировать ресурсы сети. Когда маршрутизатору предлагается обработать такой поток, он может принять или отвергнуть его, обосновывая своё решение доступной ёмкостью и количеством уже находящихся в обработке потоков [3].

Процесс принятия решения об обработке или игнорировании потока сложнее, чем простое сравнение запрашиваемых потоком параметров (пропускной способности, буферной памяти, времени цен-

трального процессора) с имеющимися. Во-первых, хотя многие приложения и знают свои требования к пропускной способности, они понятия не имеют, какой объём буферной памяти и сколько тактов работы процессора им требуется. Следовательно, нужен, по крайней мере, иной способ описания потоков. Далее, приложения весьма различаются по толерантности в отношении предельного срока обработки. Наконец, некоторые приложения могут поторговаться за параметры пакетов, а некоторые не могут. Например, проигрыватель видео, предоставляющий обычно 30 кадров/с, может согласиться работать на 25 кадрах/с, если для 30 не хватает пропускной способности. Аналогично можно настраивать количество пикселей на кадр, полосу пропускания для аудиоданных и другие свойства потоков различных приложений.

Так как на поток действуют много сторон (отправитель, приёмник и все маршрутизаторы на пути между ними), то его необходимо описывать крайне аккуратно с помощью параметров, о которых можно дискутировать. Набор таких параметров называется спецификацией потока. В типичном случае отправитель (например, сервер видеоданных) создает спецификацию потока, указывая параметры, которые он хотел бы использовать для аргументации. По мере того, как эта спецификация распространяется по пути следования потока, содержащаяся в нём информация анализируется всеми маршрутизаторами, которые модифицируют параметры так, как считают нужным. Эти модификации могут быть направлены только на снижение трафика – никто не станет сознательно брать на себя больше работы, чем требует заказчик (например, указываемая в спецификации скорость передачи данных может быть понижена, но не повышена). Когда спецификация доходит до приёмника, становятся понятны окончательные параметры.

В качестве содержимого спецификации потока рассмотрим пример, базирующийся на RFC 2210 и RFC 2211. В спецификации содержится пять параметров, первый из которых, *скорость «маркерного ведра»*, хранит число байтов, поступающих в «ведро» за секунду. Это максимум, который отправитель может поддерживать в течение довольно длительного времени, усреднённый по большому временно́му отрезку.

Второй параметр – размер «маркерного ведра» в байтах. Если, к примеру, *скорость «маркерного ведра»* составляет 1 Мбит/с, а размер ведра равен 500 кбайт, то его можно будет наполнять данными в течение 4 с. Всё, что будет посылаться после этого, будет теряться.

Третий параметр, *пиковая скорость передачи данных*, – это максимальная допустимая скорость даже для коротких промежутков времени. Отправитель ни в коем случае не должен превышать это значение.

Наконец, последние два важных параметра – минимальный и максимальный размеры пакетов, включая заголовки транспортного и сетевого уровней (например, TCP и IP). Минимальный размер важен, поскольку обработка каждого пакета занимает какое-то, пусть даже очень малое, время. Маршрутизатор может быть готов принимать 10 000 пакетов в сек. по 1 кбайт каждый, но не готов обрабатывать 100 000 пакетов по 50 байт в сек., несмотря на то, что во втором случае скорость передачи данных меньше, чем в первом. Максимальный размер пакета не менее важен, но уже по другой причине. Дело в том, что существуют определённые внутрисетевые ограничения, которые ни в коем случае не должны быть превышены. Например, если путь потока лежит через Ethernet, то максимальный размер пакета будет ограничен 1500 байтами, независимо от того, какого размера пакеты могут поддерживать другие части сети.

Маршрутизатор преобразует спецификацию потока в набор определённых резервируемых ресурсов. Это отображение является специфическим и не стандартизованным действием. Допустим, маршрутизатор может обрабатывать 100 000 пакетов/с. Если ему предлагается пропустить через себя поток со скоростью 1 Мбайт/с с максимальным размером пакета 512 байт, он может легко посчитать, что такой поток даёт 2048 пакетов/с, значит, под него необходимо отвести 2 % времени работы процессора, а лучше немного больше, чтобы избежать больших задержек обслуживания. Если политика маршрутизатора не позволяет ему резервировать более 50 % процессорного времени (что подразумевает половинную задержку) и если 49 % уже зарезервировано, то поток будет отвергнут. Подобные вычисления необходимо производить для всех резервируемых ресурсов.

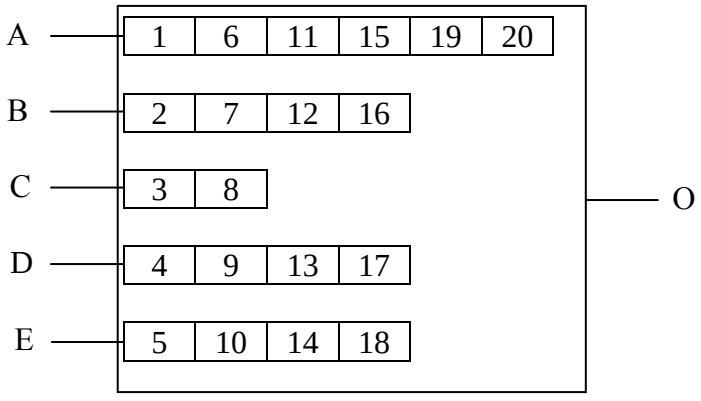
Чем строже спецификация потока данных, тем лучше для маршрутизаторов. Если же в спецификации говорится, что *скорость «маркерного ведра»* составляет 5 Мбайт/с, однако пакеты могут быть размером от 50 до 1500 байтов, значит, скорость передачи пакетов может колебаться от 3500 до 105 000 пакетов/с. Маршрутизатор может отвергнуть такой поток. При минимальном размере пакета, равном 1000 байт, 5-мегабайтный поток данных тем же самым маршрутизатором мог бы быть принят.

4.37. Пропорциональная маршрутизация

Большинство алгоритмов маршрутизации пытаются искать наилучшие пути для каждого адресата и направлять весь трафик по оптимальной траектории. Альтернативный подход, позволяющий повысить качество обслуживания, состоит в разделении трафика для одного и того же адресата между несколькими маршрутами [3]. Поскольку маршрутизаторы обычно не следят за нагрузкой на всю сеть в целом, остаётся лишь один способ разделения трафика – на основе доступной локальной информации. Одним из простых методов является маршрутизация пакетов, пропорциональная или эквивалентная емкостям исходящих связей. Однако существуют и более сложные алгоритмы [3].

4.38. Диспетчеризация пакетов

Если маршрутизатор имеет поддержку нескольких потоков, существует опасность того, что один из них захватит слишком большую часть пропускной способности и не даст жить всем остальным потокам. Обработка пакетов в порядке поступления может привести к тому, что агрессивный источник загрузит все производственные мощности маршрутизаторов, через которые проходит его поток, и тем самым снизит качество обслуживания других источников. Для пресечения подобных попыток были разработаны алгоритмы диспетчеризации пакетов (рис. 4.25) [3].



а

Пакет	Время финиширования
C	8
B	16
D	17
E	18
A	20

б

Рис. 4.25. Управление очередью пакетов:
а – маршрутизатор с пятью очередями пакетов для линии O;
б – время окончания сканирования для пяти пакетов

Одним из первых был алгоритм справедливого обслуживания. Суть его в том, что маршрутизаторы организуют отдельные очереди для каждой исходящей линии, по одной для каждого потока. Как только линия освобождается, маршрутизатор начинает циклически сканировать очереди, выбирая первый пакет следующей очереди. Таким образом, если за данную исходящую линию борются n хостов, то каждый из них имеет возможность отправить свой пакет, пропустив n - 1 чужих пакетов. Агрессивному хосту не поможет то, что в его очереди стоит больше пакетов, чем у остальных.

С этим алгоритмом связана одна проблема: предоставляемая им пропускная способность напрямую зависит от размера пакета, используемого хостом: большая часть предоставляется хостам с большими пакетами, и меньшая – хостам с небольшими пакетами. Тогда производится циклический опрос с целью выхватывания не пакета, а байта. То есть очереди сканируются побайтно до того момента, пока не будет выхвачен последний байт последнего пакета. После этого пакеты отправляются в том порядке, в котором они заканчивались при опросе очередей.

На маршрутизатор поступают пакеты длиной от 2 до 6 байтов. Во время первого такта извлекается первый байт пакета с линии *A*. Затем следует первый байт пакета с линии *B* и т. д. Первым, через 8 тактов, закончится обработка пакета *C*. Порядок сортировки пакетов соответствует времени финиширования (см. рис. 4.25, *б*). При отсутствии новых поступлений пакеты будут отсылаться в указанном порядке, начиная с *C* и заканчивая *A*.

Проблема данного алгоритма заключается в том, что он даёт всем хостам одинаковые приоритеты. Во многих случаях желательно предоставлять, например, видеосерверам, бóльшую пропускную способность, чем обычным файл-серверам, чтобы они могли посылать два или более байт за такт опроса. Такая модификация алгоритма называется взвешенным справедливым обслуживанием**.** Иногда весовой коэффициент эквивалентен числу потоков, генерируемых машиной, таким образом все процессы получают равные доли пропускной способности. Всё чаще и чаще встречаются аппаратные реализации передачи пакетов через маршрутизаторы или коммутаторы [3].

4.39. Объединение различных сетей

До сих пор неявно предполагалось наличие единой однородной сети, в которой каждая машина использует один и тот же протокол на всех уровнях. Но существует множество различных сетей, включая локальные, региональные и глобальные. Наличие различных сетей всегда приводит к возникновению различных протоколов [3].

Есть основание предполагать, что разнообразие сетей (а следовательно, и протоколов) будет оставаться всегда по следующим причинам. Прежде всего, установленная база существующих сетей уже достаточно велика и продолжает расти. Почти все персональные компьютеры используют протокол TCP/IP. Во многих больших компаниях ещё остались мейнфреймы, использующие протоколы SNA фирмы «IBM». Существенная доля телефонных сетей ориентирована на ATM. Локальные сети персональных компьютеров все еще пользуются протоколами Novell NCP/IPX или AppleTalk. Наконец, беспроводные сети – эта бурно развивающаяся сегодня область – внедряют свои протоколы. Такая тенденция будет сохраняться в ближайшие годы благодаря наличию большого количества существующих сетей и ещё благодаря тому, что некоторые производители считают, что возможность клиента легко переходить в системы других производителей не в их интересах.

Во-вторых, по мере того как компьютеры и сети становятся всё дешевле, уровень принятия решения о выборе той или иной технологии всё опускается и опускается, и теперь уже этим занимаются организации, желающие установить у себя сеть. Многие компании придерживаются политики разграничения полномочий в зависимости от стоимости технологий. При существенном удешевлении систем передачи с набором различных протоколов решение о выборе системы будет приниматься на уровне руководителей среднего и низшего звена без согласования с вышестоящими руководителями. Такая политика может легко привести к тому, что в различных отделах одной и той же организации будут установлены не согласованные между собой рабочие станции, ориентированные на несовместимые друг с другом протоколы.

В-третьих, различные сети (например, ATM и беспроводные сети) основаны на принципиально разных технологиях, поэтому вряд ли стоит удивляться, что с появлением нового оборудования появится и новое программное обеспечение для него. Например, средний дом

сейчас напоминает офис, каким он был десять лет назад: жилище битком набито разнообразными компьютерами (стационарными, ноутбуками и карманными), не соединёнными друг с другом. В будущем, возможно, будет нормой объединять в единую сеть телефон, телевизор и другую бытовую технику, так, чтобы ею можно было управлять дистанционно. Появление этой новой технологии, несомненно, повлечёт создание новых протоколов.

Рассмотрим следующий пример взаимодействия нескольких различных сетей (рис. 4.26). Показана корпоративная сеть, части которой находятся далеко друг от друга и соединены глобальной сетью ATM. В одной из частей для объединения Ethernet, беспроводной локальной сети 802.11 и мейнфреймовой сети SNA корпоративного центра данных используется оптическая магистраль FDDI.

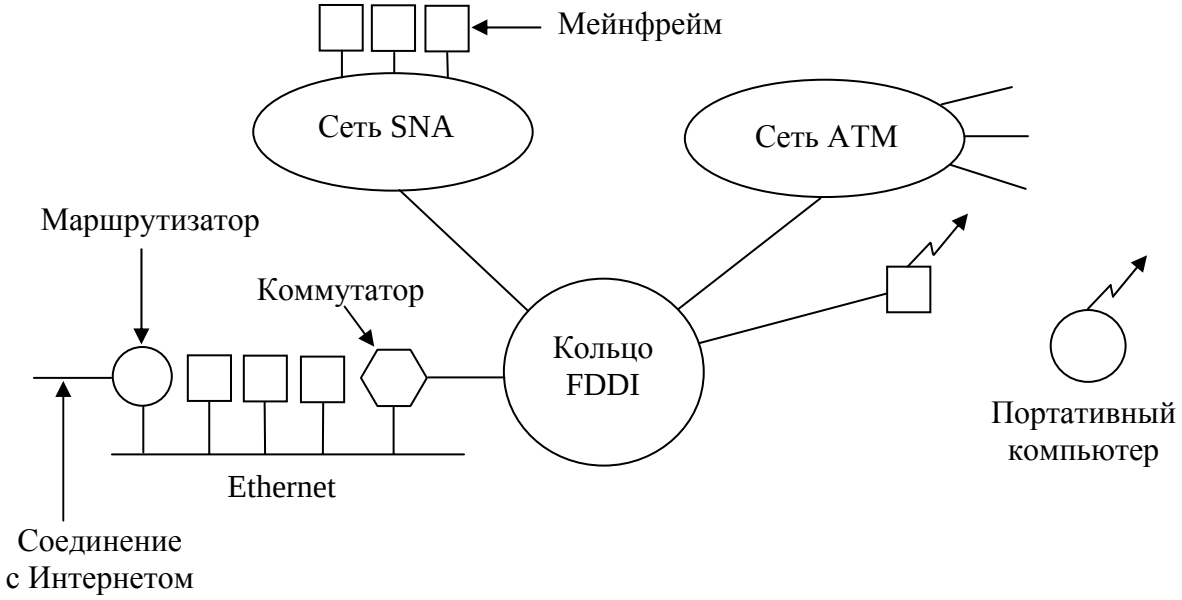


Рис. 4.26. Набор объединённых сетей

Целью объединения этих сетей является предоставление пользователям возможности общаться с пользователями любой другой из этих сетей, а также получать доступ к данным всех частей сети. Для реализации этих возможностей требуется посылать пакеты из одной сети в другую. Поскольку сети зачастую различаются довольно сильно, это действие осуществить не так уж просто.

Сети могут отличаться друг от друга довольно сильно и по разным параметрам. Некоторые из параметров, такие, как методы модуляции

или форматы кадров, нас сейчас не интересуют, поскольку они относятся к физическому уровню и уровню передачи данных. На сетевом уровне могут быть различия по следующим параметрам:

- предлагаемому сервису (ориентированные на соединение или не требующие соединения);
- протоколам (IP, IPX, SNA, ATM, MPLS, AppleTalk и др.);
- адресации (плоская (802) или иерархическая (IP));
- многоадресной рассылке (присутствует или отсутствует, а также широковещание);
- размеру пакета (у каждой сети есть свой максимум);
- качеству обслуживания (много разновидностей);
- обработке ошибок (надёжная, упорядоченная и хаотичная доставка);
- управлению потоком (скользящее окно, управление скоростью, другое или никакого);
- борьбе с перегрузкой («дырявое ведро», «маркерное ведро», сдерживающие пакеты, нерегулярное раннее обнаружение);
- безопасности (правила секретности, шифрование, кодирование, аутентификация);
- параметрам (различные тайм-ауты, спецификация потока);
- тарификации (по времени соединения, за пакет, побайтно или никак).

Именно сглаживание этих различий делает обеспечение работы объединённой сети значительно более сложным делом, чем обеспечение работы одной сети.

Когда пакетам данных приходится пересекать несколько сетей, отличных от исходной сети, может возникнуть много проблем, связанных с интерфейсами между сетями. Во-первых, когда пакеты из ориентированной на соединение сети должны пересечь не требующую соединений сеть, их порядок может нарушиться, причём для отправителя это может оказаться неожиданностью, а получатель может оказаться не подготовленным к такому событию. Часто будет требоваться преобразование протоколов, что может быть непросто, если необходимая функциональность не может быть выражена. Также понадобится преобразование форматов адресов, что может потребовать создания некой разновидности системы каталогов. Передача многоадресных пакетов через сеть, не поддерживающую многоадресную рассылку, потребует формирования отдельных пакетов для каждого адресата.

Различия в максимальном размере пакетов в разных сетях составляют главную проблему. Как передать 8000-байтовый пакет по сети, в которой максимальный размер пакета равен 1500 байтам? При передаче пакета с обязательствами доставки в реальном масштабе времени по сети, не предоставляющей каких-либо гарантий работы в реальном времени, возникает проблема разницы в качестве обслуживания.

Методы обработки ошибок, управления потоком и борьбы с перегрузкой часто различаются в разных сетях. Если отправитель и получатель ожидают, что все пакеты будут доставлены без ошибок и с сохранением порядка, а сеть просто игнорирует пакеты, когда ей угрожает перегрузка, или пакеты, направляясь различными путями, приходят к получателю совсем не в том порядке, в каком они были отправлены, то многие приложения просто не смогут работать в таких условиях. Различия в механизмах безопасности, установке параметров, правилах тарификации и даже в законах, охраняющих тайну переписки, могут послужить причиной многих проблем.

4.40. Способы объединения сетей

Сети могут объединяться с помощью разных устройств [3]. На физическом уровне сети объединяются повторителями или концентраторами, которые просто переносят биты из одной сети в другую такую же сеть. Чаще всего это аналоговые устройства, не имеющие отношения к цифровым протоколам (регенераторы сигналов).

На канальном уровне объединение идёт с помощью мостов и коммутаторов. Они могут принимать кадры, анализировать их MAC-адреса, направлять их в другие сети, осуществляя по ходу дела минимальные преобразования протоколов, например, из Ethernet в FDDI или в 802.11.

На сетевом уровне есть маршрутизаторы, соединяющие две сети. Если сетевые уровни у них разные, маршрутизатор может обеспечить перевод пакета из одного формата в другой, хотя такие преобразования сейчас выполняются всё реже. Маршрутизатор может поддерживать несколько сетевых протоколов, тогда он называется мультипротокольным маршрутизатором.

На транспортном уровне существуют транспортные шлюзы, предоставляющие интерфейсы для соединений своего уровня. Транс-

портный шлюз позволяет, к примеру, передавать пакеты из сети TCP в сеть SNA (протоколы транспортного уровня у них различаются), склеивая одно соединение с другим.

Наконец, на прикладном уровне шлюзы осуществляют преобразование семантики сообщений. Например, шлюзы между электронной почтой Интернета (RFC 822) и электронной почтой X.400 должны анализировать содержимое сообщений и изменять различные поля электронного конверта.

На рис. 4.27 показано отличие объединения на сетевом уровне от объединения на уровне передачи данных (канальном). Источник *S* пытается послать пакет приёмнику *D*. Эти две машины работают в разных сетях Ethernet, соединённых коммутатором. Источник *S* вставляет пакет в кадр и отправляет его. Кадр прибывает на коммутатор, который по MAC-адресу определяет, что его надо переслать в ЛВС-2. Коммутатор просто снимает кадр с ЛВС-1 и передаёт его в ЛВС-2.

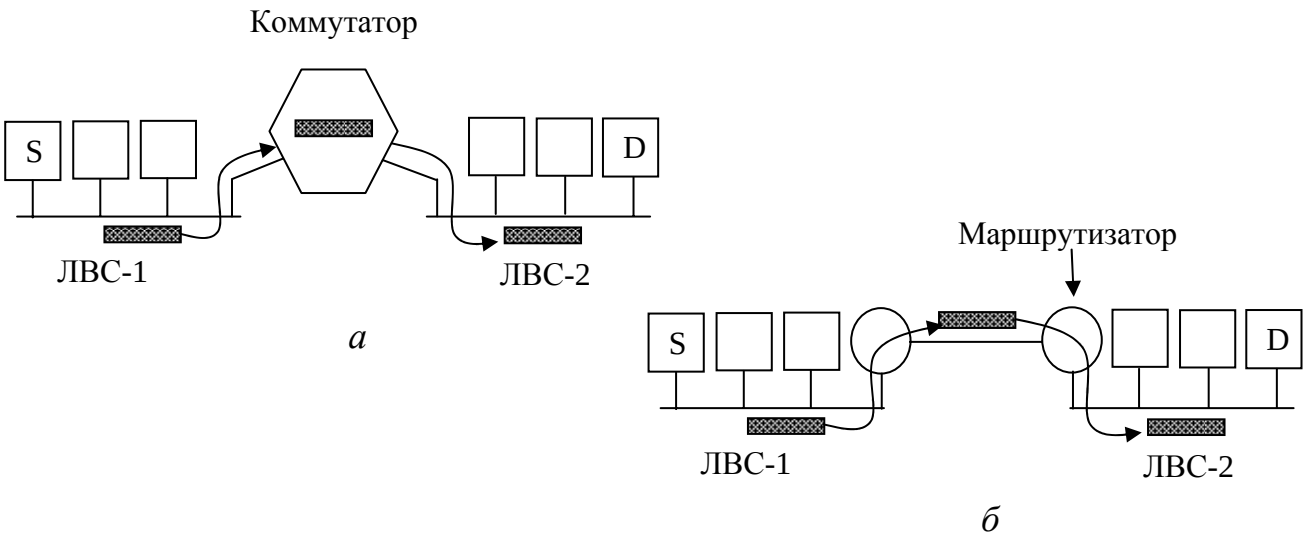


Рис. 4.27. Объединение сетей:
а – коммутатором; *б* – маршрутизатором

Теперь рассмотрим ту же ситуацию, но с применением другого сетевого оборудования. Допустим, две сети Ethernet объединены не коммутатором, а парой маршрутизаторов. Маршрутизаторы между собой соединены двухточечной линией, которая может представлять собой, например, выделенную линию длиной в тысячи километров. Что в данном случае будет происходить с кадром? Он принимается маршрутизатором, из его поля данных извлекается пакет. Далее маршрутизатор анализирует содержащийся в пакете адрес (например,

IP-адрес). Этот адрес нужно отыскать в таблице маршрутизации. В соответствии с ним принимается решение об отправке пакета (возможно, упакованного в кадр нового вида – это зависит от протокола, используемого линией) на удалённый маршрутизатор. На противоположном конце пакет вставляется в поле данных кадра Ethernet и помещается в ЛВС-2.

В чём заключается основная разница между процессами коммутации (установки моста) и маршрутизации? Коммутатор (мост) пересылает весь пакет, обосновывая своё решение значением MAC-адреса. При применении маршрутизатора пакет извлекается из кадра, и для принятия решения используется адрес, содержащийся именно в пакете. Коммутаторы не обязаны вникать в подробности устройства протокола сетевого уровня, с помощью которого производится маршрутизация, а маршрутизаторы работают именно по этому протоколу.

4.41. Сцепленные виртуальные каналы

При объединении сетей наиболее распространёнными являются два решения: ориентированное на сцепление подсетей виртуальных каналов и дейтаграммное интерсетевое объединение [3]. В прошлом большинство сетей (общего пользования) были ориентированными на соединение (сети с ретрансляцией кадров, SNA, 802.16 и ATM по сей день являются таковыми). Со стремительным развитием Интернета всё больше применялись дейтаграммы. Тем не менее было бы ошибкой думать, что дейтаграммный способ будет существовать вечно. В этом деле единственное постоянство – это изменчивость. С ростом доли и важности мультимедийных данных в общем потоке растёт вероятность того, что наступит эпоха возрождения для технологий, ориентированных на соединение. Причиной тому является тот простой факт, что при установлении соединения гораздо проще гарантировать определённый уровень обслуживания.

В модели сцепленных виртуальных каналов (рис. 4.28) соединение с хостом в удалённой сети устанавливается способом, близким к тому, как устанавливаются обычные соединения. Подсеть видит, что адресат является удалённым, и создаёт виртуальный канал к ближайшему маршрутизатору из сети адресата. Затем строится виртуальный канал от этого маршрутизатора к внешнему шлюзу (многопротокольному маршрутизатору). Этот шлюз запоминает существование созданного

виртуального канала в своих таблицах и строит новый виртуальный канал к маршрутизатору в следующей подсети. Процесс продолжается до тех пор, пока не будет достигнут хост-получатель.

Когда по проложенному пути начинают идти пакеты данных, каждый шлюз переправляет их дальше, преобразуя формат пакетов и номера виртуальных каналов. Очевидно, что все информационные пакеты будут передаваться по одному и тому же пути и, таким образом, прибудут к пункту назначения с сохранением порядка отправления.

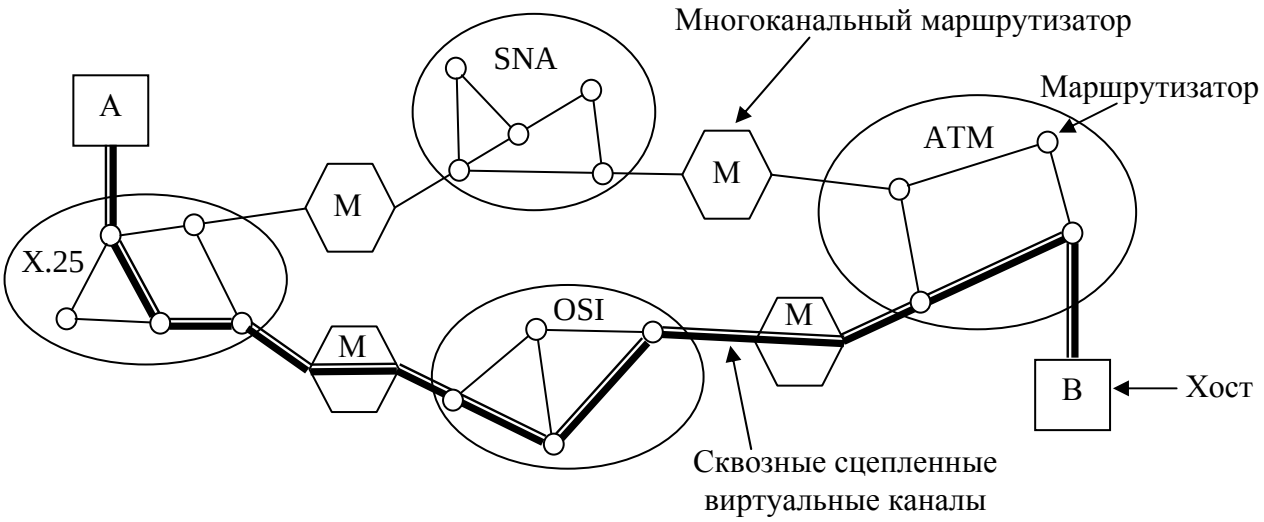


Рис. 4.28. Объединение сетей с помощью сцепленных виртуальных каналов

Существенной особенностью данного подхода является то, что последовательность виртуальных пакетов данных устанавливается от источника через один или более шлюзов к приёмнику. Каждый шлюз хранит таблицы, содержащие информацию о проходящих через них виртуальных каналах, о том, как осуществлять маршрутизацию для них и каков номер нового виртуального канала.

Такая схема работает лучше всего, когда все сети обладают примерно одинаковыми свойствами. Например, если каждая из них гарантирует надёжную доставку пакета данных сетевого уровня, то, исключив случай сбоя системы где-то на его пути, можно сказать, что и весь поток от источника до приёмника будет надёжным. С другой стороны, если машина-источник работает в сети, которая гарантирует надёжную доставку, а какая-то промежуточная сеть может терять пакеты, то сцепление радикально изменит сущность сервиса.

Сцепленные виртуальные каналы связи часто применяются на транспортном уровне. В частности, можно построить битовый канал, используя, например, SNA, который заканчивается на шлюзе, и иметь при этом TCP-соединение между соседними шлюзами. Таким образом, можно построить сквозной виртуальный канал связи, охватывающий разные сети и протоколы.

4.42. Дейтаграммное объединение сетей

Альтернативной моделью объединения сетей передачи данных является дейтаграммная модель (рис. 4.29) [3]. В данной модели единственный сервис, который сетевой уровень предоставляет транспортному уровню, состоит в возможности посылать в сеть дейтаграммы и надеяться на лучшее.

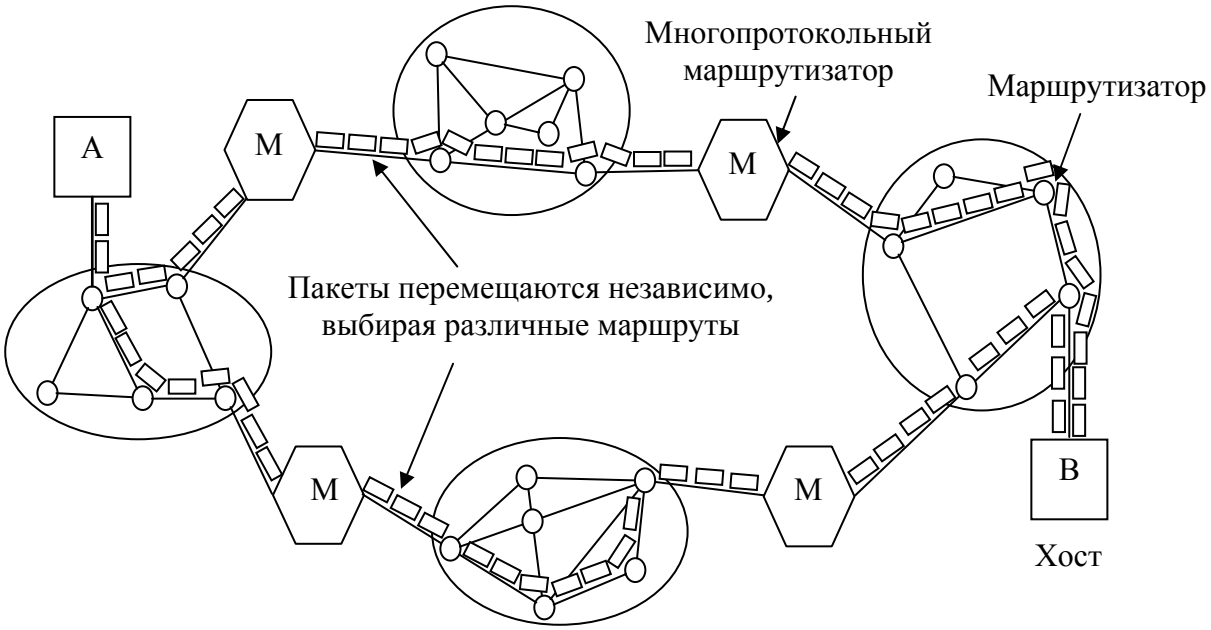


Рис. 4.29. Дейтаграммная объединённая сеть

На сетевом уровне нет никакого упоминания о виртуальных каналах, не говоря уже об их сцеплении. В этой модели пакеты не обязаны следовать по одному и тому же маршруту, даже если они принадлежат одному соединению. Дейтаграммы следуют от хоста 1 к хосту 2 с выбором различных маршрутов по объединённой сети. Выбор маршрута производится независимо для каждого пакета и может быть обусловлен текущей загруженностью сети. Могут использоваться различные маршруты, что позволяет достигать большей пропускной

способности, чем при применении модели сцепленных виртуальных каналов. С другой стороны, не даётся никакой гарантии того, что пакеты прибудут к получателю в нужном порядке, если они вообще прибудут.

Данная модель не так проста, как кажется. Если каждая сеть обладает своим сетевым уровнем, то пакет из одной сети не сможет перейти в другую сеть. Можно представить себе многопротокольные маршрутизаторы, пытающиеся преобразовать пакет из одного формата

в другой, но, если только эти форматы не являются близкими родственниками, такое преобразование всегда будет неполным и часто обречено на ошибку.

Ещё более серьёзные проблемы возникают с адресацией. Например, интернет-хост пытается послать IP-пакет другому хосту, находящемуся в соседней сети SNA. Адреса IP и SNA различаются. Значит, потребуется двухстороннее приведение одного формата адреса к другому. Более того, различается сама концепция адресации. В IP адресуемой единицей является хост (собственно, интерфейсная карта). В SNA адресуемыми могут быть не только хосты, но и другие субъекты (например, физические устройства). В идеале необходимо поддерживать специальную базу данных, отображающую одни адресуемые сущности на другие, но это задача исключительно трудоёмкая, если не сказать невыполнимая.

Другая идея заключается в выработке универсального межсетевого пакета данных, который распознавался бы всеми маршрутизаторами. Именно эта идея была взята за основу при разработке протокола IP: IP-пакеты созданы для передачи по разным сетям. Заставить всех согласиться применять только один формат практически невозможно, особенно учитывая тот факт, что каждая компания считает самым большим своим достижением изобретение, внедрение и раскручивание собственного формата.

Модель сцепленных виртуальных каналов обладает теми же преимуществами, что и применение виртуальных каналов внутри единой подсети: буферы могут быть зарезервированы заранее, сохранение порядка пакетов гарантируется, могут использоваться короткие заголовки, кроме того, можно избежать неприятностей, вызываемых дубликатами пакетов.

Недостатки опять те же самые: маршрутизаторы должны хранить таблицы с записями для каждого открытого соединения, при возник-

новении затора обходные пути не используются, а выход маршрутизатора из строя обрывает все проходящие через него виртуальные каналы. Кроме того, очень сложно, может, даже невозможно реализовать систему виртуальных каналов, если в состав объединённой сети входит хотя бы одна ненадёжная дейтаграммная сеть.

Свойства дейтаграммного подхода к объединению сетей те же самые, что и у дейтаграммных подсетей: риск возникновения перегрузки выше, но также больше и возможностей для адаптации к ней, высока надёжность в случае отказов маршрутизаторов, однако требуются более длинные заголовки пакетов. В объединённой сети, как и в единой дейтаграммной сети, возможно применение различных адаптивных алгоритмов выбора маршрута.

Главное преимущество дейтаграммного подхода к объединению сетей заключается в том, что он может применяться в подсетях без виртуальных каналов. К дейтаграммным сетям относятся многие локальные, мобильные (в том числе применяемые в воздушном и морском транспорте) и даже некоторые глобальные сети. При включении одной из этих сетей в объединённую сеть стратегия объединения сетей на основе виртуальных каналов встречает серьёзные трудности.

Выводы по главе 4

На сетевом уровне решается задача разработки маршрутов доставки пакетов от отправителя к получателю. Здесь требуется создание алгоритмов оптимальной маршрутизации с использованием многих транзитных участков сети. Другая задача уровня – корректное сопряжение пользователей, находящихся в различных сетях.

На этом уровне основным действующим звеном сети выступает маршрутизатор, связанный каналами с такими же устройствами. Стоит задача эффективного управления этим звеном.

Сетевой уровень предоставляет транспортному уровню сервисы с установлением и без установления соединения. Причём основная функция сетевого уровня заключается в выборе маршрута для пакетов от начальной до конечной точки. В большинстве сетей пакетам приходится проходить через несколько маршрутизаторов. Алгоритм сетевой маршрутизации должен отвечать требованиям оптимальности с точки зрения скорости доставки сообщений, равномерности загруз-

ки каналов, гибкости реакции на изменения трафика во времени, а также обладать устойчивостью.

Задача оптимальности решается исходя из конкретной обстановки по составу и поведению сети и выбранного критерия. Сетевые алгоритмы выбора маршрута могут быть адаптивными и неадаптивными. Первые изменяют решение о выборе маршрутов при изменении топологии сети и также часто – в зависимости от загруженности линий. Вторые не учитывают при выборе маршрута топологию и текущее состояние сети и не изменяют трафик на линиях. Выбор маршрута для каждой пары пользователей производится заранее, в автономном режиме (статическая маршрутизация).

Метод лавинной маршрутизации позволяет наилучшим образом обеспечить гарантированность доставки пакетов сообщения адресату. Но часто данный метод не является оптимальным, поскольку происходит лавинообразное размножение пакетов на параллельных ветвях сети. Сеть перегружается циркулирующей информацией с многократным дублированием. И даже после фактического получения адресатом всего сообщения в сети продолжают перемещаться пакеты данных, создавая перегрузку и высокую вероятность ошибочного повторного приёма информации (одна из разновидностей сетевых атак). Следовательно, необходим контроль прохождения информации через каждый маршрутизатор путём регистрации номеров (или иных меток) пакетов, что требует наличия в нём элементов памяти. Для устранения упомянутых недостатков применяется выборочная заливка в конкретных приложениях.

Маршрутизация по вектору расстояний и с учётом состояния линии позволяет оптимально решать задачу доставки сообщений. Но требуется применять «интеллектуальные» сетевые маршрутизаторы, способные самостоятельно, по заданной программе, анализировать ситуацию в каналах связи, поддерживать взаимодействие с соседями (другими маршрутизаторами) и принимать решение о переключениях или блокировании сетевого трафика. Для взаимодействия с соседями маршрутизатор должен создавать пакеты служебной информации, где содержится информация о необходимом оповещении и постановка текущих задач по обработке конкретного пакета.

В больших, сложных, распределённых сетях целесообразно вводить иерархическую маршрутизацию, чтобы избежать проблем, связанных с увеличением памяти для таблиц маршрутов и времени обработки всей этой служебной информации процессором. В определён-

ный момент сеть может вырасти до таких размеров, при которых перестанет быть возможным хранение на маршрутизаторах записи обо всех остальных маршрутизаторах. Поэтому в больших сетях маршрутизация пакетов должна осуществляться иерархически. Маршрутизаторы разбиваются на отдельные регионы, которые определяются им как зоны ответственности. В своём регионе каждый маршрутизатор хранит в памяти все детали подсети, освобождаясь от необходимости знать подробности топологии соседних зон.

Многоадресная рассылка предполагает выбор оптимальных, наиболее коротких путей и информирование об этом всех маршрутизаторов, которые предполагается задействовать на данной траектории.

Если участники информационного обмена находятся в движении, это ведёт к дополнительным трудностям маршрутизации. Мобильные хосты привносят новое усложнение в и без того непростое дело выбора маршрутов в различных вычислительных сетях – чтобы направить пакет к мобильному хосту, его нужно сначала найти. Здесь актуален вопрос о средствах и возможностях привязки мобильных абонентов к точкам доступа стационарной сети и, разумеется, о механизме их адресации. Следует учитывать также множество дестабилизирующих факторов, связанных с различными условиями и физическими средами распространения информационных сигналов.

Для выбора оптимального маршрута пакетов узлы коммутации направляют запросы по сети широковещательным способом. Запросы должны иметь стандартизированный формат, чтобы они были понятны устройствам различных подсетей. Готовые к работе узлы (маршрутизаторы) должны сообщать в ответ свои состояния и наличие связей. Для принятия решения важен временно́й интервал исполнения следующей части цикла управления: запрос – сбор информации – ответ. Одновременно должно быть ограничено время жизни каждого такого пакета служебной информации, поскольку она объективно быстро устаревает.

Производительность сети может снижаться из-за перегрузки – превышения количеством одновременно циркулирующих пакетов некоего порогового уровня. Наличие перегрузки означает, что нагрузка временно превысила возможности ресурсов данной части системы. В особо напряжённые моменты может иметь место ситуация полной блокировки сети по причине невозможности обеспечения информационного обмена. Известно, что одновременное обращение нескольких объектов к одному ресурсу ведёт к конфликтным ситуациям. То-

гда в сети начинается не только задержка, но и потеря пакетов данных. Когда число пакетов достигает максимального уровня, производительность сети начинает снижаться. При очень высоком уровне трафика производительность сети падает до совсем низкого уровня и практически никакие пакеты не доставляются. Причины перегрузки: ненормативное (превышающее расчётное) увеличение трафика, недостаточная память для буферизации пакетов, недостатки программного обеспечения, не предусматривающего ограничение времени жизни пакетов и тайм-ауты на их повторную передачу, медленные процессоры, некорректная работа интерфейсов, которые не способны согласовать скорости и форматы взаимодействия различных элементов сети.

В главе рассмотрены различные принципы, стратегии, методы и алгоритмы борьбы с перегрузками. Здесь следует отметить, что применение комбинаций этих методов и алгоритмов может дать значительно больший эффект, чем применение только одного из них. В любом случае, решение принимается исходя из принципов целесообразности и разумной достаточности.

Вопросы для самопроверки

1. Задачи, решаемые на сетевом уровне ВОС.
2. Виды коммутации пакетов и их характеристики.
3. В чём заключается понятие оптимальности маршрута?
4. В чем состоит основное различие между ЛВС и ГВС?
5. Принципы выбора кратчайшего пути пакета.
6. Что такое вектор расстояний?
7. Поясните смысл понятия «идентификатор соединений».
8. Базовые требования к маршрутизаторам.
9. Что такое иерархическая маршрутизация?
10. Как организуется широковещательная и многоадресная рассылка?
11. Опишите процедуру регистрации в сети для мобильных объектов.
12. Форматы пакетов запроса маршрута и наличия маршрута.
13. Причины возникновения перегрузки в сети.
14. Перечислите методы борьбы с перегрузками.
15. Алгоритмы «дырявого ведра» и «маркерного ведра».
16. Как происходит диспетчеризация пакетов?
17. Проблемы взаимодействия сетей при их объединении.
18. Необходимость и способы объединения сетей.

ЛИТЕРАТУРА

1. Ирвин, Д. Передача данных в сетях: инженерный подход / Д. Ирвин, Д. Харль. – СПб.: БХВ-Петербург, 2003. – 448 с.

2. Куроуз, Дж. Компьютерные сети / Дж. Куроуз, К. Росс. – 2-е изд. – СПб.: Питер, 2004. – 765 с.

3. Олифер, В.Г. Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов / В.Г. Олифер, Н.А. Олифер. – 4-е изд. – СПб.: Питер, 2010. – 944 с.

4. Парк, Дж. Передача данных в системах контроля и управления: практическое руководство / Дж. Парк, С. Маккей, Э. Райт. – М.: ООО «Группа ИДТ», 2007. – 480 с.

5. Першин, В.Т. Основы современной радиоэлектроники: учебное пособие / В.Т. Першин. – Ростов н/Д: Феникс, 2009. – 541 с.

6. Пескова, С.А. Сети и телекоммуникации: учебное пособие для студентов высших учебных заведений / С.А. Пескова, А.В. Кузин, А.Н. Волков. – М.: Изд-кий центр «Академия», 2009. – 352 с.

7. Таненбаум, Э. Компьютерные сети / Э. Таненбаум. – 4-е изд. – СПб.: Питер, 2003. – 992 с.

8. Хогдал, Дж. Скотт Анализ и диагностика компьютерных сетей / Дж. Скотт Хогдал. – М.: Издательство «Лори», 2001. – 362 с.

Еременко Владимир Тарасович
Лобанова Валентина Андреевна
Тютякин Александр Васильевич
Георгиевский Александр Евгеньевич
Донцов Венедикт Михайлович
Воронина Оксана Александровна

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ. ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ (Часть 1)

Конспект лекций

Редактор Г.В. Карпушина
Технический редактор Т.П. Прокудина

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Государственный университет - учебно-научно-производственный комплекс»
Лицензия ИД № 00670 от 05.01.2000 г.

Подписано к печати 17.04.2012 г. Формат 60x84 1/16.
Усл. печ. л. 20,8. Тираж 100 экз.
Заказ №_____

Отпечатано с готового оригинал-макета
на полиграфической базе ФГБОУ ВПО «Госуниверситет - УНПК»,
302030, г. Орел, ул. Московская, 65.