МИНИСТЕРСТВО ОБАЗОВАНИЯ И НАУКИ, МОЛОДЕЖИ И СПОРТА УКРАИНЫ
ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

КОНСПЕКТ ЛЕКЦИЙ

по курсу

"Компьютерные сети"

Для студентов, обучающихся по направлению
6.050903 “Телекомуникации”
(для дневной и заочной форм обучения)

Рассмотрено
на заседании кафедры
автоматики и телекомуникаций
Протокол № 4 от 12.04.2012г.

Утверждено на заседании учебно-
издательского совета ДонНТУ
Протокол № 2 от 19.04.2012г.

Донецк, ДонНТУ 2012 р.

Конспект лекций по курсу "Компьютерные сети" (для студентов направления подготовки 6.050903 “Телекомуникации” (ТКС) дневной и заочной форм обучения)/ Составители: Р.В. Федюн, В.А. Попов- Донецк: ДонНТУ, 2012.- 218 с.

Составители: Р.В. Федюн, доц.
В.А. Попов, доц.

Рецензент В.А. Светличная, доц.
В.В. Червинский, доц.

Ответственный за выпуск В.Я.Воропаева, зав. каф.

1. ЭВОЛЮЦИЯ КОМПЬЮТЕРНЫХ СИСТЕМ И СЕТЕЙ

Сети передачи данных, называемые также вычислительными или компьютерными сетями, являются результатом эволюции двух важнейших научно-технических отраслей современной цивилизации – компьютерных и телекоммуникационных технологий (рис. 1.1).

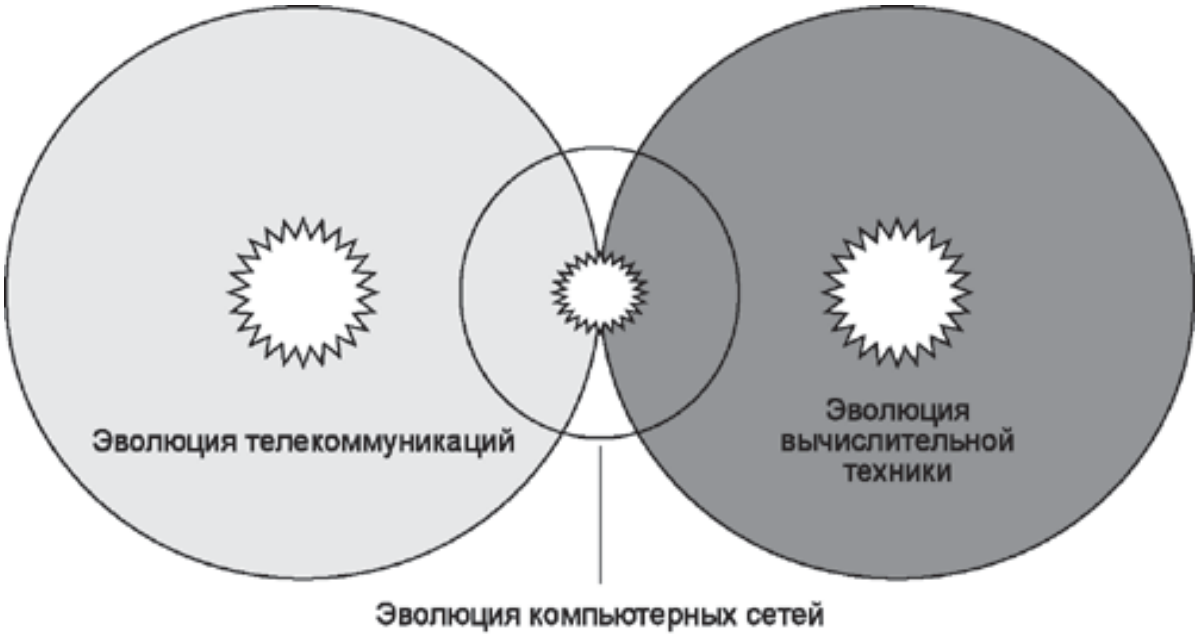


Рисунок 1.1 Эволюция компьютерных сетей на стыке вычислительной техники и телекоммуникационных технологий

С одной стороны, сети передачи данных представляют собой частный случай распределенных вычислительных систем, в которых группа компьютеров согласованно выполняет набор взаимосвязанных задач, обмениваясь данными в автоматическом режиме; с другой – компьютерные сети могут рассматриваться как средство передачи информации на большие расстояния, для чего в них применяются методы кодирования и мультиплексирования данных, получившие развитие в различных телекоммуникационных системах. Итак, ***компьютерная сеть*** – это набор компьютеров, связанных коммуникационной системой и снабженных соответствующим программным обеспечением, которое предоставляет пользователям сети доступ к ресурсам этого набора компьютеров.

Идея компьютера была предложена английским математиком Чарльзом Бэбиджем (Charles Babig) в середине девятнадцатого века. Однако его механическая "аналитическая машина" по-настоящему так и не заработала.

Подлинное рождение цифровых вычислительных машин произошло вскоре после окончания второй мировой войны. В середине 40-х годов XX века были созданы первые ламповые вычислительные устройства. Для этого периода характерно следующее:

- компьютер представлял собой скорее предмет исследования, а не инструмент для решения каких-либо практических задач из других областей;

- программирование осуществлялось исключительно на машинном языке;

- не было никакого системного программного обеспечения, кроме библиотек математических и служебных подпрограмм;

- операционные системы еще не появились, все задачи организации вычислительного процесса решались вручную каждым программистом с пульта управления.

С середины 50-х годов XX века начался следующий период в развитии вычислительной техники, связанный с появлением новой технической базы – полупроводниковых элементов. В этот период:

− выросло быстродействие процессоров, увеличились объемы оперативной и внешней памяти;

− появились первые алгоритмические языки, и, таким образом, к библиотекам математических и служебных подпрограмм добавился новый тип системного программного обеспечения – трансляторы;

− разработаны первые системные управляющие программы – мониторы, которые автоматизировали всю последовательность действий оператора по организации вычислительного процесса.

Программные мониторы явились прообразом современных операционных систем, они стали первыми системными программами, предназначенными не для обработки данных, а для управления вычислительным процессом.

1.1. Мультипрограммирование

Следующий важный период развития операционных систем относится к 1965 – 1975 годам. В это время в технической базе вычислительных машин произошел переход от отдельных полупроводниковых элементов типа транзисторов к интегральным микросхемам, что открыло путь к появлению следующего поколения компьютеров, представителем которого является, например, IBM/360.

В этот период были реализованы практически все основные механизмы, присущие современным операционным системам (ОС): мультипрограммирование, мультипроцессирование, поддержка многотерминального многопользовательского режима, виртуальная память, файловые системы, разграничение доступа и сетевая работа. В эти годы начинается расцвет системного программирования. Из направления прикладной математики, представляющего интерес для узкого круга специалистов, системное программирование превращается в отрасль индустрии, оказывающую непосредственное влияние на практическую деятельность миллионов людей.

В условиях резко возросших возможностей компьютера, связанных с обработкой и хранением данных, выполнение только одной программы в каждый момент времени оказалось крайне неэффективным.

Начались разработки в области мультипрограммирования.

Мультипрограммирование – способ организации вычислительного процесса, при котором в памяти компьютера находится одновременно несколько программ, попеременно выполняющихся на одном процессоре.

Мультипрограммирование было реализовано в двух вариантах:

• пакетная обработка;

• разделение времени.

Системы пакетной обработки предназначались для решения задач в основном вычислительного характера, не требующих быстрого получения результатов. Главной целью и критерием эффективности систем пакетной

обработки является максимальная пропускная способность, то есть решение максимального числа задач в единицу времени.

Для достижения этой цели в системах пакетной обработки используется следующая схема функционирования (рис. 1.2): в начале работы формируется пакет заданий, каждое задание содержит требование к системным ресурсам; из этого пакета заданий формируется мультипрограммный набор, то есть множество одновременно выполняемых задач. Для одновременного выполнения выбираются задачи, предъявляющие к ресурсам различные требования, так, чтобы обеспечивалась сбалансированная загрузка всех устройств вычислительной машины. Например, в мультипрограммном наборе желательно присутствие и вычислительных задач, и задач с интенсивным вводом-выводом. Таким образом, выбор нового задания из пакета заданий зависит от внутренней ситуации, складывающейся в системе, то есть выбирается "выгодное" задание. Следовательно, в вычислительных системах, работающих под управлением пакетных ОС, невозможно гарантировать выполнение того или иного задания в течение определенного периода времени.

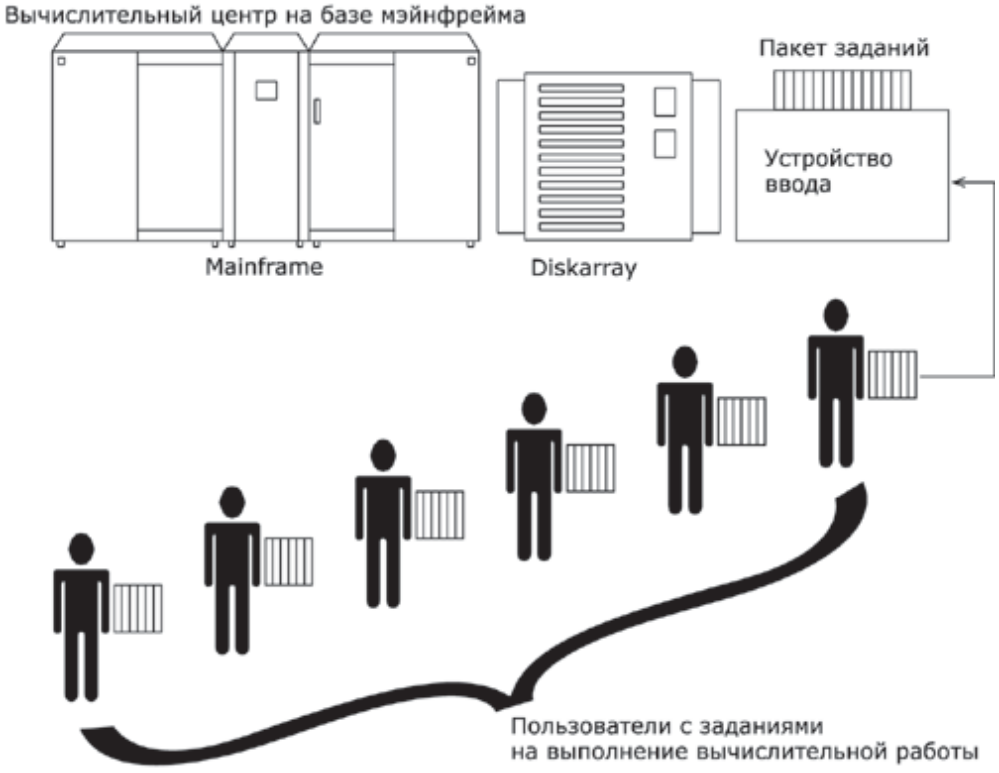


Рисунок 1.2. Централизованный характер вычислений в системах пакетной обработки

В системах пакетной обработки переключение процессора с одной задачи на другую происходит по инициативе самой активной задачи, например, когда она "отказывается" от процессора из-за необходимости выполнить операцию ввода-вывода. Поэтому существует высокая вероятность того, что одна задача может надолго занять процессор, и выполнение интерактивных задач станет невозможным. Взаимодействие пользователя с вычислительной машиной, на которой установлена система пакетной обработки, сводится к тому, что пользователь приносит задание, отдает его диспетчеру-оператору, а в конце дня после выполнения всего пакета заданий получает результат. Очевидно, что такой порядок повышает эффективность функционирования аппаратуры, но снижает эффективность работы пользователя.

В **системах разделения времени** пользователям (или одному пользователю) предоставляется возможность интерактивной работы сразу с несколькими приложениями. Для этого каждое приложение должно регулярно взаимодействовать с пользователем. Понятно, что в пакетных системах возможности диалога пользователя с приложением ограничены.

В системах разделения времени эта проблема решается за счет того, что ОС принудительно периодически приостанавливает приложения, не дожидаясь, когда они сами освободят процессор. Всем приложениям попеременно выделяется квант процессорного времени. Таким образом, пользователи, запустившие программы на выполнение, получают возможность поддерживать с ними диалог.

Системы разделения времени призваны исправить основной недостаток систем пакетной обработки – изоляцию пользователя-программиста от процесса выполнения задач. Каждому пользователю в этом случае предоставляется терминал, с которого он может вести диалог со своей программой. Так как в системах разделения времени каждой задаче выделяется только квант процессорного времени, ни одна задача не занимает процессор надолго, и время ответа оказывается приемлемым. Если квант небольшой, то у всех пользователей, одновременно работающих на одной и той же машине, складывается впечатление, что каждый из них использует машину единолично.

1.2.Многотерминальные системы – прообраз сети

Терминалы, выйдя за пределы вычислительного центра, рассредоточились по всему предприятию. Многотерминальный режим использовался не только в системах разделения времени, но и в системах пакетной обработки. При этом не только оператор, но и все пользователи получали возможность формировать свои задания и управлять их выполнением со своего терминала. Такие операционные системы получили название систем удаленного ввода заданий.

Терминальные комплексы могли располагаться на большом расстоянии от процессорных стоек, соединяясь с ними с помощью различных глобальных связей – модемных соединений телефонных сетей или выделенных каналов. Для поддержки удаленной работы терминалов в операционных системах появились специальные программные модули, реализующие различные (в то время, как правило, нестандартные) протоколы связи. Такие вычислительные системы с удаленными терминалами, сохраняя централизованный характер обработки данных, в какой-то степени являлись прообразом современных компьютерных сетей (рис.1.3), а соответствующее системное программное обеспечение – прообразом сетевых операционных систем.

Многотерминальные централизованные системы уже имели все внешние признаки локальных вычислительных сетей, однако по существу ими не являлись, так как сохраняли сущность централизованной обработки данных автономно работающего компьютера.

Действительно, рядовой пользователь работу за терминалом мэйнфрейма воспринимал примерно так же, как сейчас воспринимает работу за подключенным к сети персональным компьютером. Пользователь мог получить доступ к общим файлам и периферийным устройствам, при этом у него создавалась полная иллюзия единоличного владения компьютером, так как он мог запустить нужную ему программу в любой момент и почти сразу же получить результат.

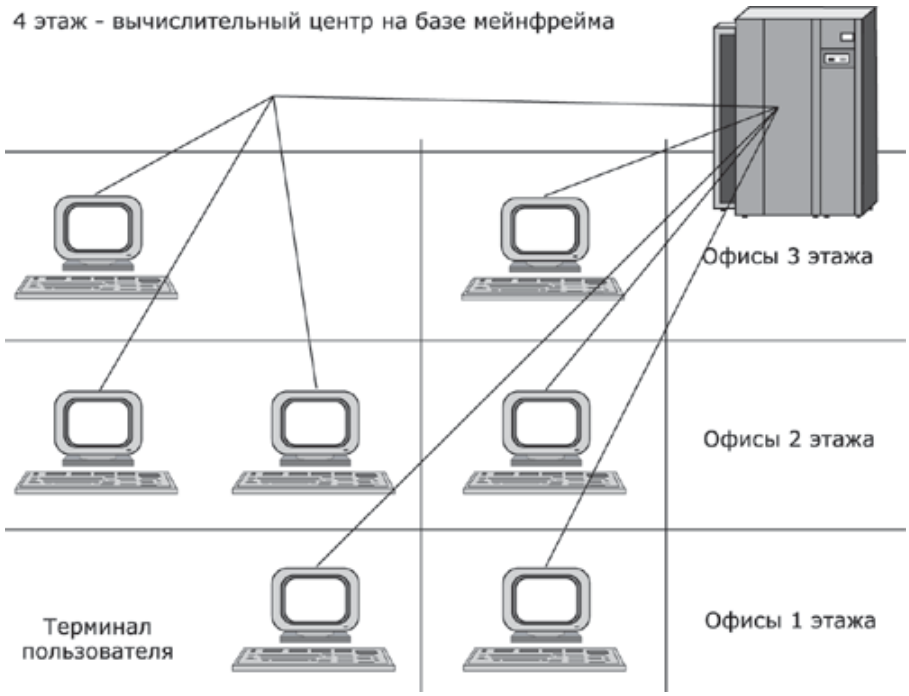


Рисунок 1.3. Многотерминальная система – прообраз вычислительной сети

1.3.Первые сети – глобальные

Хотя теоретические работы по созданию концепций сетевого взаимодействия велись почти с момента появления вычислительных машин, значимые практические результаты по объединению компьютеров в сети были получены лишь в конце 60-х, когда с помощью глобальных связей и техники коммутации пакетов удалось реализовать взаимодействие машин класса мэйнфреймов и суперкомпьютеров (рис.1.4). Эти дорогостоящие компьютеры хранили уникальные данные и программы, обмен которыми позволил повысить эффективность их использования.

Рисунок 1.4. Объединение удаленных супер-ЭВМ глобальными связями

Но еще до реализации связей "компьютер-компьютер" была решена более простая задача – организация связи "удаленный терминал-компьютер". Терминалы, находящиеся от компьютера на расстоянии многих сотен, а то и тысяч километров, соединялись с компьютерами через телефонные сети с помощью модемов. Такие сети позволяли многочисленным пользователям получать удаленный доступ к разделяемым ресурсам нескольких мощных компьютеров класса супер-ЭВМ.

И только потом были разработаны средства обмена данными между компьютерами в автоматическом режиме. На основе этого механизма в первых сетях были реализованы службы обмена файлами, синхронизации баз данных, электронной почты и другие, ставшие теперь традиционными, сетевые службы.

В 1969 году министерство обороны США инициировало работы по объединению в общую сеть суперкомпьютеров оборонных и научно-исследовательских центров. Эта сеть, получившая название ARPANET послужила отправной точкой для создания первой и самой известной ныне глобальной сети – Internet. Сеть ARPANET объединяла компьютеры разных типов, работавшие под управлением различных ОС с дополнительными модулями, реализующими коммуникационные протоколы, общие для всех компьютеров сети. Такие ОС можно считать первыми сетевыми операционными системами.

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

В 1974 году компания IBM объявила о создании собственной сетевой архитектуры для своих мэйнфреймов, получившей название SNA (System Network Architecture, системная сетевая архитектура). В это же время в Европе активно велись работы по созданию и стандартизации сетей X.25.

Таким образом, хронологически первыми появились глобальные сети (Wide Area Networks, WAN), то есть сети, объединяющие территориально рассредоточенные компьютеры, возможно, находящиеся в различных городах и странах. Именно при построении глобальных сетей были впервые предложены и отработаны многие основные идеи и концепции современных вычислительных сетей, такие, например, как многоуровневое построение коммуникационных протоколов, технология коммутации пакетов и маршрутизация пакетов в составных сетях.

Развитие технологии глобальных компьютерных сетей во многом определялся прогрессом телефонных сетей. С конца 60-х годов в телефонных сетях все чаще стала применяться передача голоса в цифровой форме, что привело к появлению высокоскоростных цифровых каналов, соединяющих АТС и позволяющих одновременно передавать десятки и сотни разговоров. Была разработана специальная технология плезиохронной цифровой иерархии (Plesiochronous Digital Hierarchy, PDH), предназначенная для создания так называемых первичных, или опорных, сетей. Такие сети не предоставляют услуг конечным пользователям, они являются фундаментом, на котором строятся скоростные цифровые каналы "точка-точка", соединяющие оборудование другой (так называемой наложенной) сети, которая уже работает на конечного пользователя.

Первоначально технология PDH, поддерживающая скорости до 140 Мбит/с, была внутренней технологией телефонных компаний. Однако со временем эти компании стали сдавать часть своих каналов PDH в аренду предприятиям, которые использовали их для создания собственных телефонных и глобальных компьютерных сетей.

Появившаяся в конце 80-х годов технология синхронной цифровой иерархии (Synchronous Digital Hierarchy, SDH) расширила диапазон скоростей цифровых каналов до 10 Гбит/с, а технология спектрального мультиплексирования DWDM (Dense Wave Division Multiplexing) – до сотен гигабит и даже нескольких терабит в секунду.

1.4. Мини-компьютеры – предвестники локальных сетей

В начале 70-х годов произошло важное событие, непосредственно повлиявшее на эволюцию компьютерных сетей.

В результате технологического прорыва в области производства компьютерных компонентов появились большие интегральные схемы (БИС). Их сравнительно невысокая стоимость и богатые функциональные возможности привели к созданию мини-компьютеров, которые стали реальными конкурентами мэйнфреймов.

Даже небольшие подразделения предприятий получили возможность иметь собственные компьютеры. К середине 70-х годов стали широко использоваться мини-компьютеры PDP-11, Nova, HP.

С помощью мини-компьютеров осуществлялось управление технологическим оборудованием и выполнялись другие задачи уровня отдела предприятия. Таким образом, появилась концепция распределения компьютерных ресурсов по всему предприятию. Однако при этом все компьютеры одной организации по-прежнему продолжали работать автономно (рис.1.5).

Архитектура мини-компьютеров была значительно упрощена по сравнению с мэйнфреймами, что нашло отражение и в их операционных системах. Многие функции мультипрограммных многопользовательских ОС мэйнфреймов были усечены, с учетом ограниченности ресурсов мини-компьютеров. Операционные системы мини-компьютеров часто стали делать специализированными, например, только для управления в реальном времени (ОС RT-11 для мини-компьютеров PDP-11) или только для поддержания режима разделения времени (RSX-11M для

тех же компьютеров). Эти операционные системы не всегда были многопользовательскими, что во многих случаях оправдывалось невысокой стоимостью машин. Важной вехой в истории мини-компьютеров и вообще в истории операционных систем стало создание ОС Unix

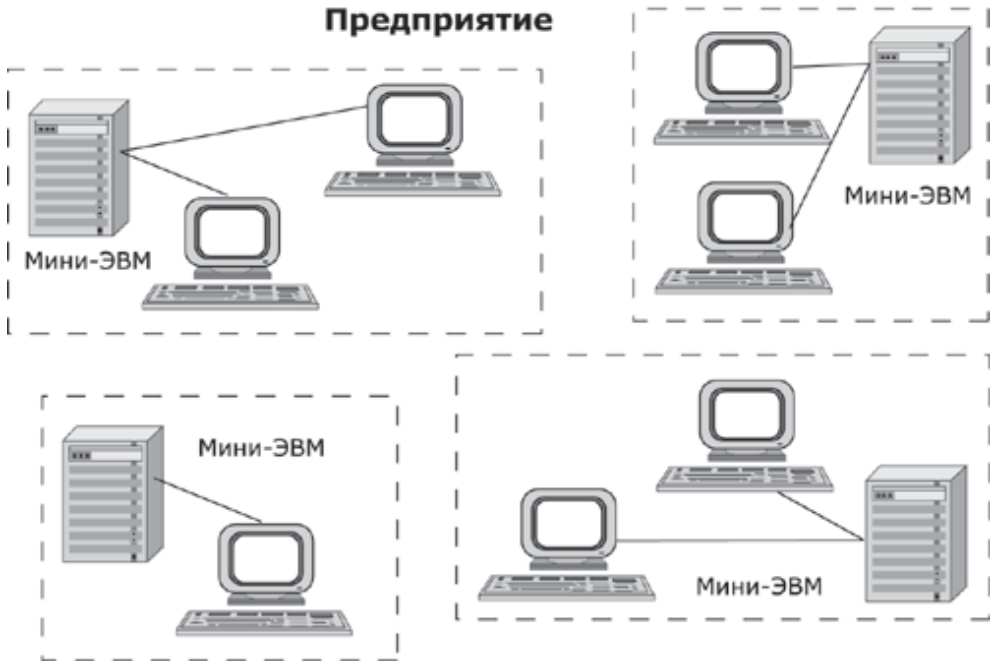


Рисунок 1.5. Автономное использование нескольких мини-компьютеров на одном предприятии

1.5. Появление стандартных технологий локальных сетей

В середине 80-х годов XX века положение дел в локальных сетях стало меняться. Утвердились стандартные технологии объединения компьютеров в сеть – Ethernet, Arcnet, Token Ring, Token Bus, несколько позже – FDDI.

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

Стандартные сетевые технологии сделали задачу построения локальной сети почти тривиальной. Для создания сети достаточно было приобрести сетевые

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

В 80-е годы были приняты основные стандарты на коммуникационные технологии для локальных сетей: в 1980 году – Ethernet, в 1985 – Token Ring, в конце 80-х – FDDI. Это позволило обеспечить совместимость сетевых операционных систем на нижних уровнях, а также стандартизировать интерфейс ОС с драйверами сетевых адаптеров.

Конец 90-х выявил явного лидера среди технологий локальных сетей – семейство Ethernet, в которое вошли классическая технология Ethernet 10 Мбит/с, а также Fast Ethernet 100 Мбит/с и Gigabit Ethernet 1000 Мбит/с. Простые алгоритмы работы предопределили низкую стоимость оборудования Ethernet. Широкий диапазон иерархии скоростей позволяет рационально строить локальную сеть, применяя ту технологию, которая в наибольшей степени отвечает задачам предприятия и потребностям пользователей. Важно также, что все технологии Ethernet очень близки друг другу по принципам работы, что упрощает обслуживание и интеграцию построенных на их основе сетей.

2. ОСНОВНЫЕ ПРОБЛЕМЫ ПОСТРОЕНИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ

При создании вычислительных сетей разработчикам пришлось решать множество самых разных задач, связанных с кодированием и синхронизацией электрических (оптических) сигналов, выбором конфигурации физических и логических связей, разработкой схем адресации устройств, созданием различных способов коммутации, мультиплексированием и демультиплексированием потоков данных, совместным использованием передающей среды.

Начнем с наиболее простого случая непосредственного соединения двух устройств физическим каналом, такое соединение называется связью "точка-точка" **(**point-to-point**)**.

2.1. Связь компьютера с периферийными устройствами

Частным случаем связи "точка-точка" является соединение компьютера с периферийным устройством. Поскольку механизмы взаимодействия компьютеров в сети многое позаимствовали у схемы взаимодействия компьютера с периферийными устройствами, начнем рассматривать принципы работы сети с этого "досетевого" случая.

Для обмена данными компьютер и периферийное устройство (ПУ) оснащены внешними интерфейсами или портами (рис. 2.1). В данном случае к понятию "интерфейс" относятся:

• электрический разъем;

• набор проводов, соединяющих устройства;

• совокупность правил обмена информацией по этим проводам.

Со стороны компьютера логикой передачи сигналов на внешний интерфейс управляют:

• ***контроллер ПУ*** – аппаратный блок, часто реализуемый в виде отдельной платы;

• ***драйвер ПУ*** – программа, управляющая контроллером периферийного устройства.

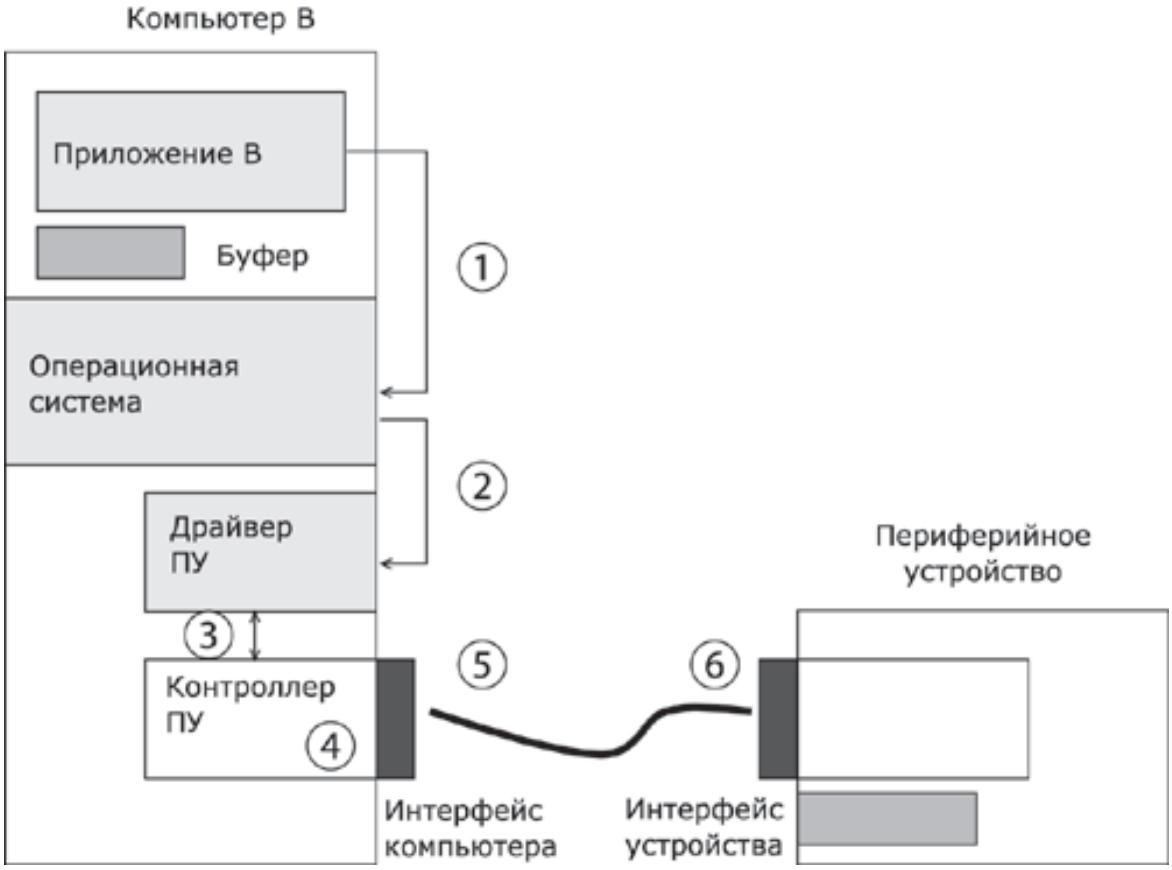


Рисунок 2.1. Связь компьютера с периферийным устройством

Со стороны ПУ интерфейс чаще всего реализуется аппаратным устройством управления ПУ, хотя встречаются и программно-управляемые периферийные устройства.

Обмен данными между ПУ и компьютером, как правило, является двунаправленным. Так, например, даже принтер, который представляет собой устройство вывода информации, возвращает в компьютер данные о своем состоянии.

Таким образом, по каналу, связывающему внешние интерфейсы, передается следующая информация:

− данные, поступающие от контроллера на ПУ, например байты текста, который нужно распечатать на бумаге;

− команды управления, которые контроллер передает на устройство управления ПУ; в ответ на них оно выполняет специальные действия, например переводит головку диска на соответствующую дорожку или же выталкивает из принтера лист бумаги;

− данные, возвращаемые устройством управления ПУ в ответ на запрос от контроллера, например данные о готовности к выполнению операции.

Рассмотрим последовательность действий, которые выполняются в том случае, когда некоторому приложению требуется напечатать текст на принтере.

1. Приложение обращается с запросом на выполнение операции печати к операционной системе. В запросе указываются: адрес данных в оперативной памяти, идентифицирующая информация принтера и операция, которую требуется выполнить (например, чтение или запись).

2. Получив запрос, операционная система анализирует его, решает, может ли он быть выполнен, и если решение положительное, то запускает соответствующий драйвер, передавая ему в качестве параметров адрес выводимых данных. Дальнейшие действия, относящиеся к операции ввода-вывода, со стороны компьютера реализуются совместно драйвером и контроллером принтера.

3. Драйвер передает команды и данные контроллеру, который помещает их в свой внутренний буфер.

4. Контроллер перемещает данные из внутреннего буфера во внешний порт.

5. Контроллер начинает последовательно передавать биты в линию связи, представляя каждый бит соответствующим электрическим сигналом. Чтобы сообщить устройству управления принтера о том, что начинается передача байта, перед передачей первого бита данных контроллер формирует стартовый сигнал специфической формы, а после передачи последнего информационного бита – стоповый сигнал. Эти сигналы синхронизируют передачу байта. Кроме информационных бит, контроллер может передавать бит контроля четности для повышения достоверности обмена.

6. Устройство управления принтера, обнаружив на соответствующей линии стартовый бит, выполняет подготовительные действия и начинает принимать информационные биты, формируя из них байт в своем приемном буфере. Если передача сопровождается битом четности, то выполняется проверка корректности передачи: при правильно выполненной передаче в соответствующем регистре устройства управления принтера устанавливается признак завершения приема информации. Наконец, принятый байт обрабатывается принтером – выполняется соответствующая команда или печатается символ.

Возможное распределение функций между драйвером и контроллером (УУ).

Функции, выполняемые драйвером:

− ведение очередей запросов;

− буферизация данных;

− подсчет контрольной суммы последовательности байтов;

− анализ состояния ПУ;

− загрузка очередного байта данных (или команды) в регистр контроллера;

− считывание байта данных или байта состояния ПУ из регистра контроллера.

Функции, выполняемые контроллером:

− преобразование байта из регистра (порта) в последовательность бит;

− передача каждого бита в линию связи;

− обрамление байта стартовым и стоповым битами – синхронизация;

− формирование бита четности;

− установка признака завершения приема/передачи байта.

2.2. Связь двух компьютеров

Предположим, что пользователь другого компьютера хотел бы распечатать текст. Сложность состоит в том, что к его компьютеру не подсоединен принтер и требуется воспользоваться тем принтером, который связан с другим компьютером (рис. 2.2).

Программа, работающая на одном компьютере, не может получить непосредственный доступ к ресурсам другого компьютера – его дискам, файлам, принтеру. Она может только "попросить" об этом другую программу, выполняемую на том компьютере, которому принадлежат эти ресурсы. Эти "просьбы" выражаются в виде сообщений, передаваемых по каналам связи между компьютерами. Такая организация печати называется удаленной.

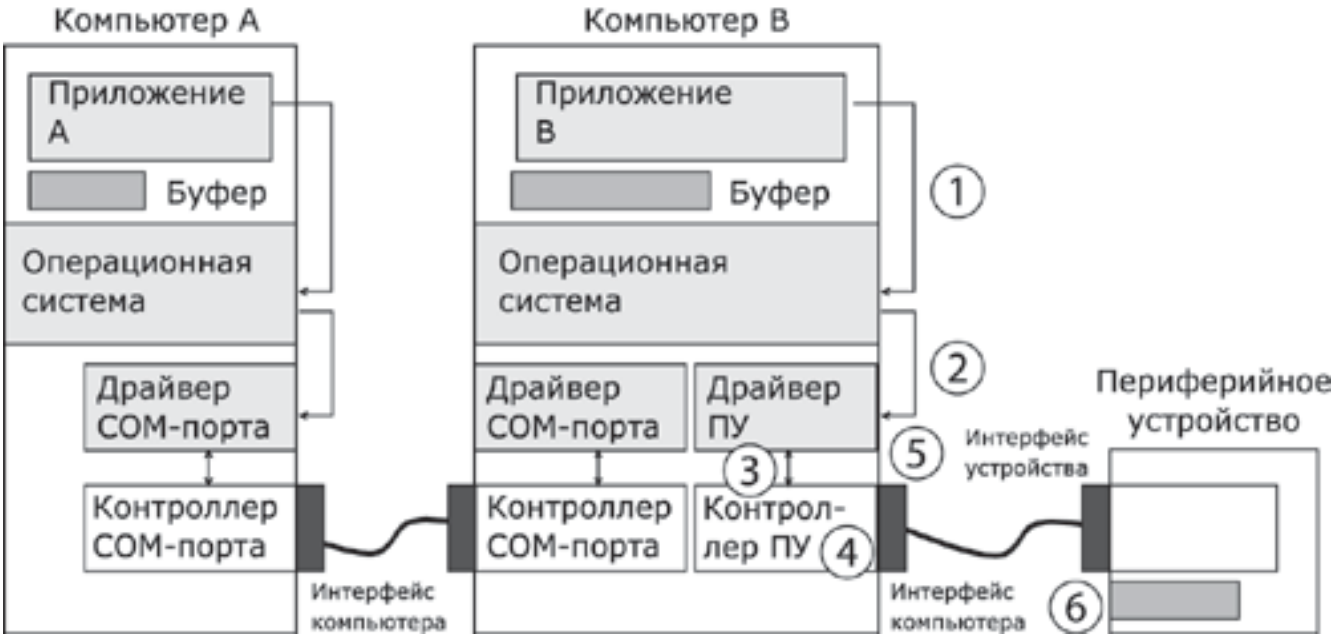


Рисунок 2.2. Взаимодействие двух компьютеров

Предположим, что мы связали компьютеры по кабелю через COM-порты, которые, как известно, реализуют интерфейс RS-232C (такое соединение часто называют нуль-модемным). Связь между компьютерами осуществляется аналогично связи компьютера с ПУ. Только теперь контроллеры и драйверы портов действуют с двух сторон. Вместе они обеспечивают передачу по кабелю между компьютерами одного байта информации.

Итак, механизм обмена байтами между двумя компьютерами определен. Теперь нужно договориться о правилах обмена сообщениями между приложениями А и В. Приложение В должно "уметь" расшифровать получаемую от приложения А информацию. Для этого программисты, разрабатывавшие

приложения А и В, строго оговаривают форматы сообщений, которыми будут обмениваться приложения, и их семантику.

Вернемся к последовательности действий, которые необходимо выполнить для распечатки текста на принтере "чужого" компьютера.

1. Приложение А формирует очередное сообщение (содержащее, например, строку, которую необходимо вывести на принтер) приложению В, помещает его в буфер оперативной памяти и обращается к ОС с запросом на передачу содержимого буфера на компьютер В.

2. ОС компьютера А обращается к драйверу COM-порта, который инициирует работу контроллера.

3. Действующие с обеих сторон пары драйверов и контроллеров COM-порта последовательно, байт за байтом, передают сообщение на компьютер В.

4. Драйвер компьютера В периодически выполняет проверку на наличие признака завершения приема, устанавливаемого контроллером при правильно выполненной передаче данных, и при его появлении считывает принятый байт из буфера контроллера в оперативную память, тем самым делая его доступным для программ компьютера В. В некоторых случаях драйвер вызывается асинхронно, по прерываниям от контроллера. Аналогично реализуется и передача байта в другую сторону – от компьютера В к компьютеру А.

5. Приложение В принимает сообщение, интерпретирует его, и в зависимости от того, что в нем содержится, формирует запрос к своей ОС на выполнение тех или иных действий с принтером. В нашем примере сообщение содержит указание на печать текста, поэтому ОС передает драйверу принтера запрос на печать строки.

Далее выполняются все действия 1–5, описывающие выполнение запроса приложения к ПУ в соответствии с рассмотренной ранее схемой "локальная ОС – драйвер ПУ – контроллер ПУ – устройство управления ПУ" (см. предыдущий раздел). В результате строка будет напечатана.

Рассмотрели последовательность работы системы при передаче только одного сообщения от приложения А к приложению В. Однако порядок

взаимодействия этих двух приложений может предполагать неоднократный обмен сообщениями разного типа. Например, после успешной печати строки (в предыдущем примере) согласно правилам, приложение В должно послать сообщение-подтверждение.

Это ответное сообщение приложение В помещает в буферную область оперативной памяти, а далее с помощью драйвера COM-порта передает его по каналу связи в компьютер А, где оно и попадает к приложению А.

2.3. Клиент, редиректор и сервер

Можно представить, что любая программа, которой потребуется печать на "чужом" принтере, должна включать в себя функции, подобные тем, которые выполняет приложение А. Но нагружать этими стандартными действиями каждое приложение – текстовые и графические редакторы, системы управления базами данных и другие приложения – не очень рационально (хотя существует большое количество программ, которые действительно самостоятельно решают все задачи по обмену данными между компьютерами, например Kermit – программа обмена файлами через COM-порты, реализованная для различных ОС, Norton Commander 3.0 с его функцией Link). Гораздо выгоднее создать специальный программный модуль, который (вместо приложения А) будет выполнять формирование сообщений-запросов к удаленной машине и прием результатов для всех приложений. Такой служебный модуль называется ***клиентом***.

На стороне же компьютера В (на месте приложения В) должна работать другая специализированная программа – ***сервер***, постоянно ожидающий прихода запросов на удаленный доступ к принтеру (или файлам, расположенным на диске) этого компьютера. Схема взаимодействия клиента и сервера с приложениями и локальной операционной системой приведена на рис. 2.3.

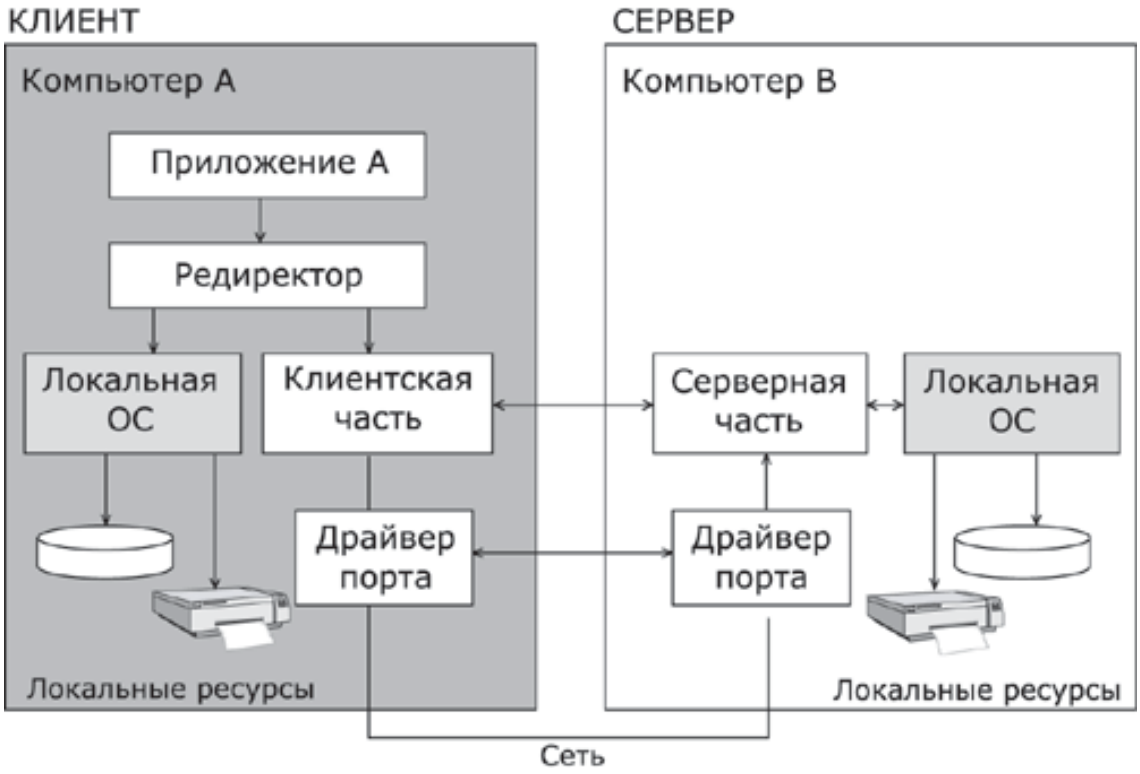


Рисунок 2.3. Взаимодействие программных компонентов при связи двух компьютеров

Для того чтобы компьютер мог работать в сети, его операционная система должна быть дополнена клиентским и/или серверным модулем, а также средствами передачи данных между компьютерами. В результате такого добавления операционная система компьютера становится ***сетевой ОС***.

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

- выбор топологии;
- адресация узлов в сети;
- доступ к общей среде передачи.

3. ТОПОЛОГИЯ ФИЗИЧЕСКИХ СВЯЗЕЙ

3.1. Типы конфигураций связи компьютеров

Как только компьютеров становится больше двух, возникает проблема выбора ***конфигурации физических связей*** или ***топологии***. Под топологией сети понимается конфигурация графа, вершинам которого соответствуют конечные узлы сети (например, компьютеры) и коммуникационное оборудование (например, маршрутизаторы), а ребрам – электрические и информационные связи между ними.

Число возможных конфигураций резко возрастает при увеличении числа связываемых устройств. Так, если три компьютера мы можем связать двумя способами, то для четырех компьютеров (рис. 3.1) можно предложить уже шесть топологически различных конфигураций (при условии неразличимости компьютеров).

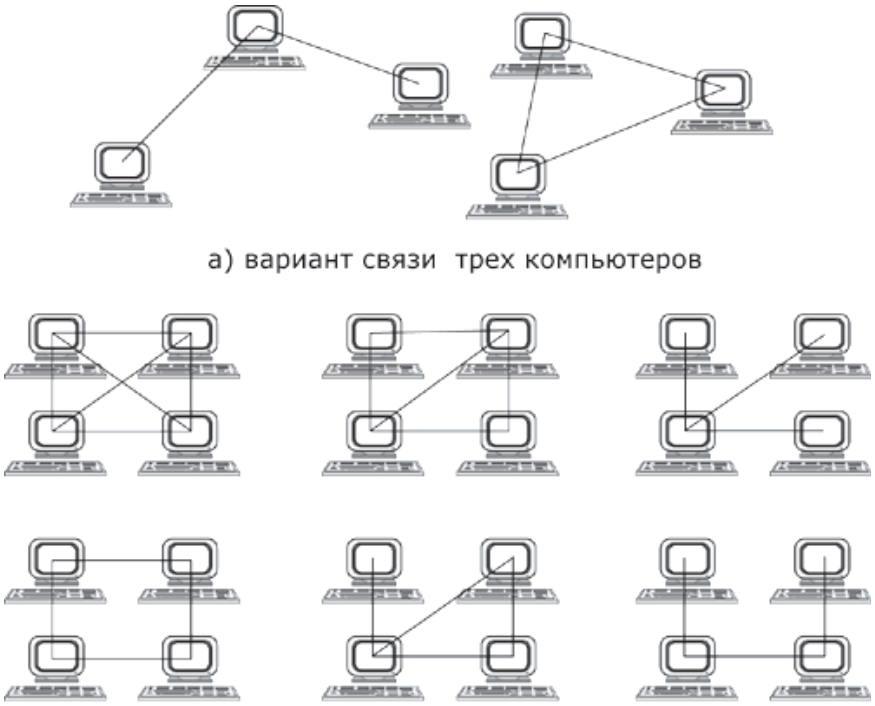


Рисунок 3.1. Варианты связи компьютеров:

а – трех компьютеров; *б* – четырех компьютеров

Среди множества возможных конфигураций различают ***полносвязные*** и ***неполносвязные*** (см. рис.3.2).

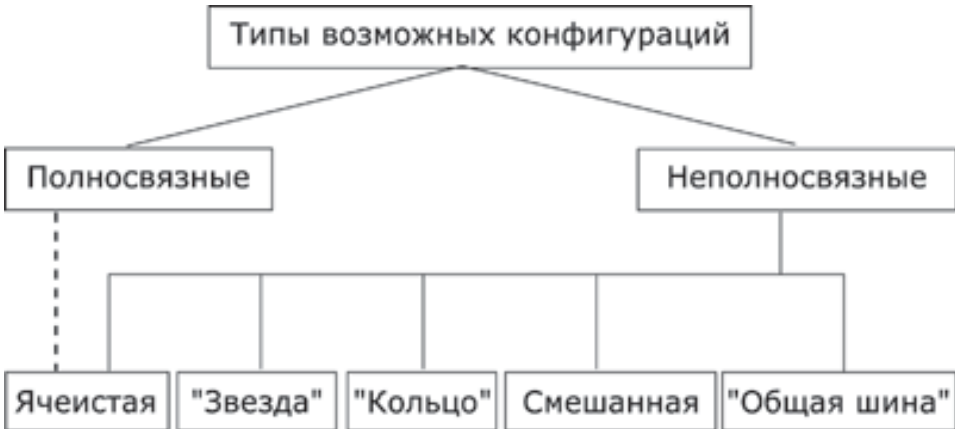


Рисунок 3.2. Типы конфигураций

Полносвязная топология (рис. 3.3) соответствует сети, в которой каждый компьютер непосредственно связан со всеми остальными. Несмотря на логическую простоту, этот вариант громоздкий и неэффективный. Действительно, каждый компьютер в сети должен иметь большое количество коммуникационных портов, достаточное для связи с каждым из остальных компьютеров. Для каждой пары компьютеров должна быть выделена отдельная физическая линия связи. (Полносвязные топологии в крупных сетях применяются редко, так как для связи N узлов требуется N(N-1)/2 физических дуплексных линий связи, то есть имеет место квадратическая зависимость. Чаще этот вид топологии используется в многомашинных комплексах или в сетях, объединяющих небольшое количество компьютеров.

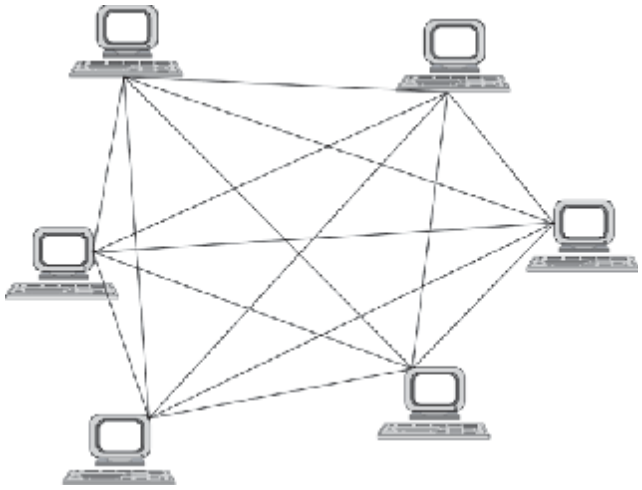


Рисунок 3.3. Полносвязная конфигурация

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

Ячеистая топология (mesh) получается из полносвязной путем удаления некоторых возможных связей. Ячеистая топология допускает соединение большого количества компьютеров и характерна для крупных сетей (рис. 3.4).

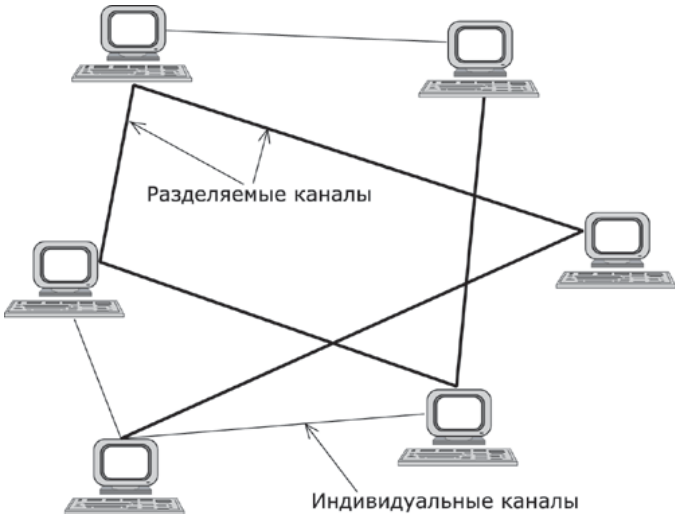


Рисунок 3.4. Ячеистая топология

В сетях с кольцевой конфигурацией (рис. 3.5) данные передаются по кольцу от одного компьютера к другому.

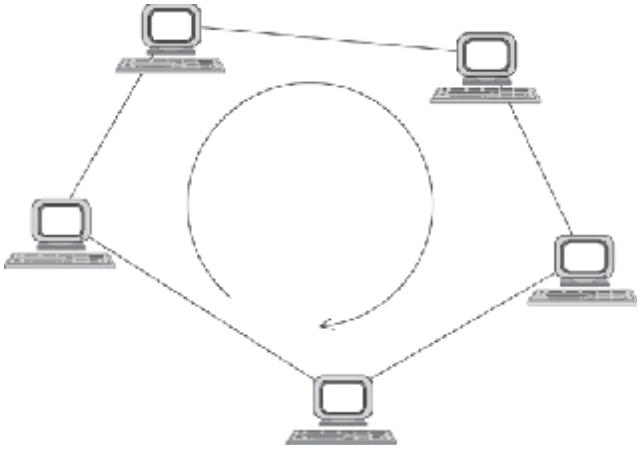


Рисунок 3.5. Топология "кольцо"

Главное достоинство "кольца" в том, что оно по своей природе обладает свойством резервирования связей. Действительно, любая пара узлов соединена здесь двумя путями – по часовой стрелке и против. "Кольцо" представляет собой

очень удобную конфигурацию и для организации обратной связи – данные, сделав полный оборот, возвращаются к узлу-источнику. Поэтому отправитель в указанном случае может контролировать процесс доставки данных адресату. Часто это свойство "кольца" используется для тестирования связности сети и поиска узла, работающего некорректно. В то же время в сетях с кольцевой топологией необходимо принимать специальные меры, чтобы в случае выхода из строя или отключения какой-либо станции не прерывался канал связи между остальными станциями "кольца".

Топология "звезда" (рис. 3.6) образуется в том случае, когда каждый компьютер с помощью отдельного кабеля подключается к общему центральному устройству, называемому концентратором. В функции концентратора входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети. В роли концентратора может выступать как компьютер, так и специализированное устройство, такое как многовходовый повторитель, коммутатор или маршрутизатор.

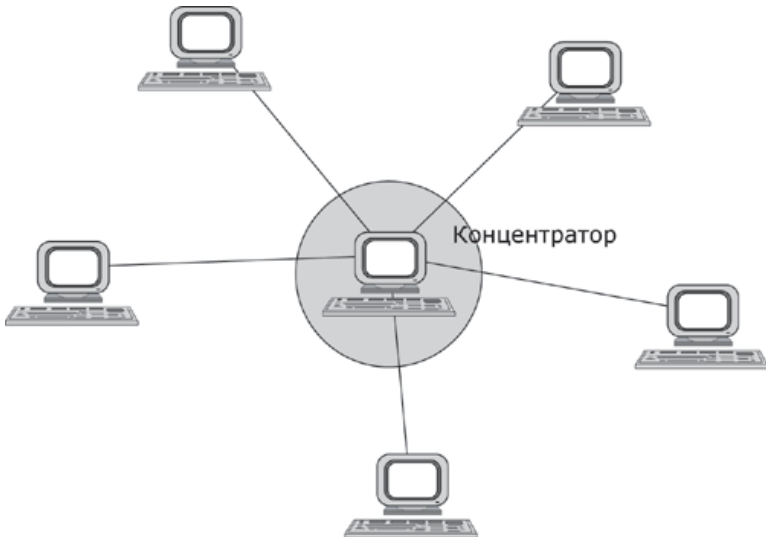


Рисунок 3.6. Топология "звезда"

К недостаткам топологии типа "звезда" относится более высокая стоимость сетевого оборудования, связанная с необходимостью приобретения специализированного центрального устройства. Кроме того, возможности наращивания количества узлов в сети ограничиваются количеством портов концентратора.

Иногда имеет смысл строить сеть с использованием нескольких концентраторов, иерархически соединенных между собой связями типа "звезда" (рис. 3.7). Получаемую в результате структуру называют также **деревом.** В настоящее время дерево является самым распространенным типом топологии связей, как в локальных, так и в глобальных сетях.

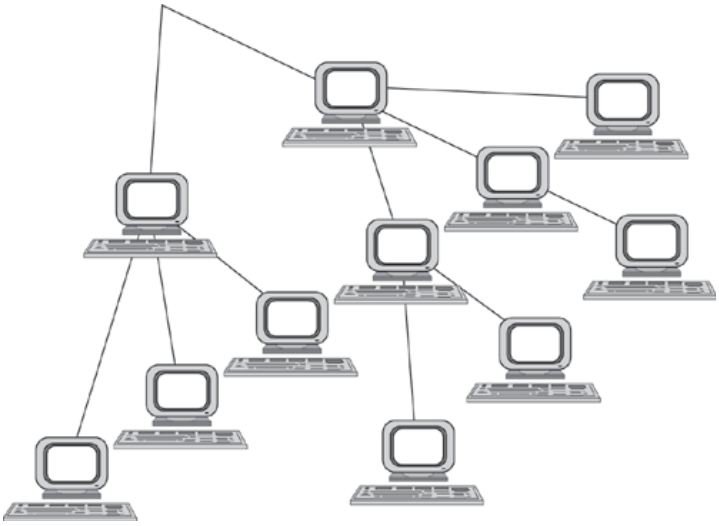


Рисунок 3.7. Топология "иерархическая звезда" или "дерево"

Особым частным случаем конфигурации "звезда" является конфигурация "общая шина" (рис. 3.8). Здесь в роли центрального элемента выступает пассивный кабель, к которому по схеме "монтажного ИЛИ" подключается несколько компьютеров (такую же топологию имеют многие сети, использующие беспроводную связь – роль общей шины здесь играет общая радиосреда).

Рисунок 3.8. Топология "общая шина"

Передаваемая информация распространяется по кабелю и доступна одновременно всем присоединенным к нему компьютерам.

Основными преимуществами такой схемы являются низкая стоимость и простота наращивания, то есть присоединения новых узлов к сети.

Самым серьезным недостатком "общей шины" является ее недостаточная надежность: любой дефект кабеля или какого-нибудь из многочисленных разъемов полностью парализует всю сеть. Другой недостаток "общей шины" – невысокая производительность, так как при таком способе подключения в каждый момент времени только один компьютер может передавать данные по сети, поэтому пропускная способность канала связи всегда делится между всеми узлами сети. До недавнего времени "общая шина" являлась одной из самых популярных топологий для локальных сетей.

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

4. АДРЕСАЦИЯ УЗЛОВ СЕТИ

Еще одной проблемой, которую нужно учитывать при объединении трех и более компьютеров, является проблема их адресации, точнее адресации их сетевых интерфейсов

По количеству адресуемых интерфейсов адреса можно классифицировать следующим образом:

- **уникальный адрес** (unicast) используется для идентификации отдельных интерфейсов;

- **групповой адрес** (multicast) идентифицирует сразу несколько интерфейсов, поэтому данные, помеченные групповым адресом, доставляются каждому из узлов, входящих в группу;

- данные, направленные по **широковещательному адресу** (broadcast), должны быть доставлены всем узлам сети;

- в новой версии протокола IPv6 определен **адрес произвольной рассылки** (anycast), который, так же как и групповой адрес, задает группу адресов, однако данные, посланные по этому адресу, должны быть доставлены не всем адресам данной группы, а любому из них.

Адреса могут быть:

- аппаратными (MAC-адрес 00.la.ff.ff);

- числовыми (IP-адрес 129.26.255.255);

- символьными (site.domen.ru, willi-winki).

Символьные адреса (имена) предназначены для запоминания людьми и поэтому обычно несут смысловую нагрузку.

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

Адресное пространство может иметь плоскую (линейную) организацию (рис. 4.1) или иерархическую организацию (рис. 4.2).

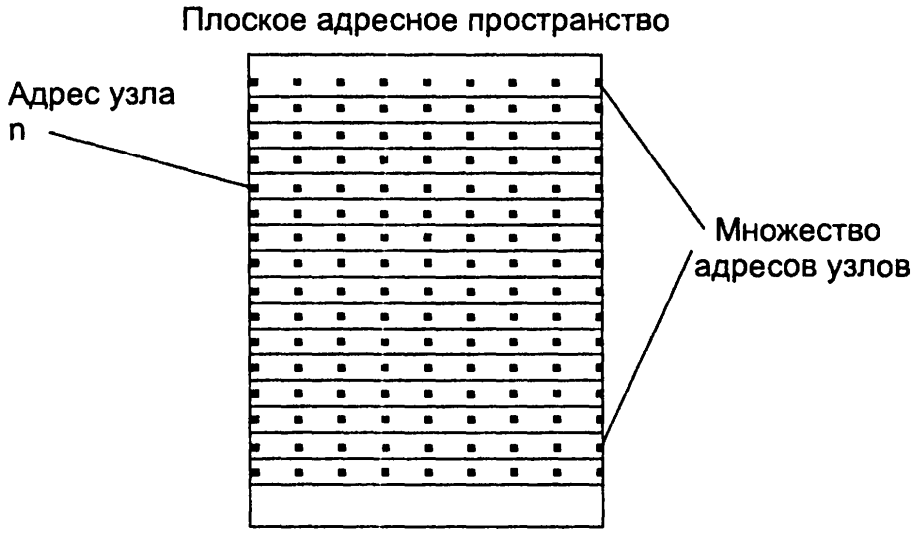


Рисунок 4.1. Плоская организация адресного пространства

При **плоской организации** множество адресов никак не структурировано. Примером плоского числового адреса является **MAC-адрес**. Такой адрес обычно используется только аппаратурой, поэтому его стараются сделать по возможности компактным и записывают в виде двоичного или шестнадцатеричного числа, например 00-la-ff-ff. При задании MAC-адресов не требуется выполнение ручной работы, так как они обычно встраиваются в аппаратуру компанией-изготовителем, поэтому их называют также **аппаратными адресами** (hardware addresses).

Рисунок. 4.2. Иерархическая организация адресного пространства

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

В показанной на рис. 4.2 трехуровневой структуре адресного пространства адрес конечного узла задается тремя составляющими: идентификатором группы (K), в которую входит данный узел, идентификатором подгруппы (L) и, наконец, идентификатором узла (n), однозначно определяющим его в подгруппе. Иерархическая адресация во многих случаях оказывается более рациональной, чем плоская. В больших сетях, состоящих из многих тысяч узлов, использование плоских адресов приводит к большим издержкам — конечным узлам и коммуникационному оборудованию приходится оперировать таблицами адресов, состоящими из тысяч записей. В противоположность этому иерархическая система адресации позволяет при перемещении данных до определенного момента пользоваться только старшей составляющей адреса (например, идентификатором группы K), затем для дальнейшей локализации адресата задействовать следующую по старшинству часть (L) и в конечном счете — младшую часть (n).

Типичными представителями иерархических числовых адресов являются сетевые IP- и IPX-адреса. В них поддерживается двухуровневая иерархия, адрес делится на старшую часть — номер сети и младшую — номер узла.

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

5. ПРОТОКОЛ. ИНТЕРФЕЙС. СТЕК ПРОТОКОЛОВ

Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе обмена сообщениями участвуют две стороны, то есть в данном случае необходимо организовать согласованную работу двух "иерархий", работающих на разных компьютерах. Оба участника сетевого обмена должны принять множество соглашений. Например, они должны согласовать уровни и форму электрических сигналов, способ определения длины сообщений, договориться о методах контроля достоверности и т.п. Другими словами, соглашения должны быть приняты для всех уровней, начиная от самого низкого – уровня передачи битов – до самого высокого, реализующего сервис для пользователей сети.

На рис. 5.1 показана модель взаимодействия двух узлов. С каждой стороны средства взаимодействия представлены четырьмя уровнями. Процедура взаимодействия этих двух узлов может быть описана в виде набора правил взаимодействия каждой пары соответствующих уровней обеих участвующих сторон.

Формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах, называются **протоколом**.

Модули, реализующие протоколы соседних уровней и находящиеся в одном узле, также взаимодействуют друг с другом в соответствии с четко определенными правилами с помощью стандартизированных форматов сообщений. Эти правила принято называть **интерфейсом**.

Интерфейс – определяет последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на соседних уровнях в одном узле. Интерфейс определяет набор услуг, предоставляемый данным уровнем соседнему уровню.

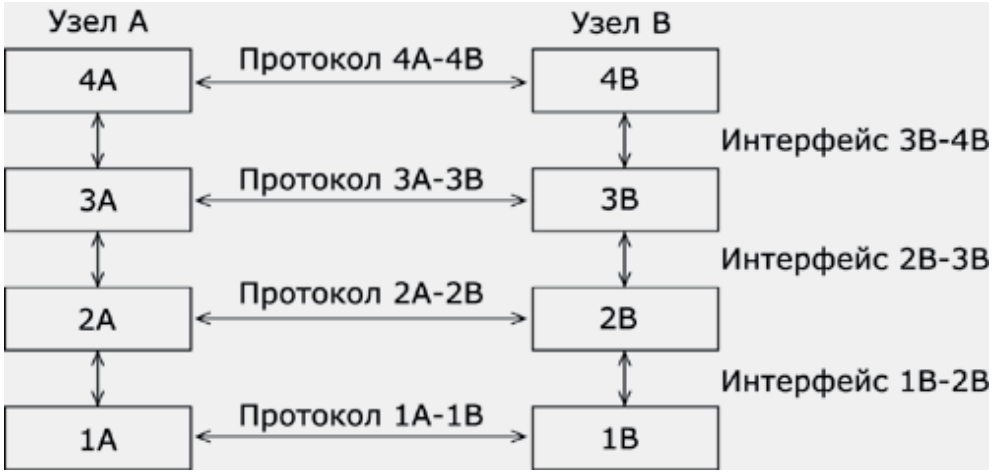


Рисунок 5.1. Взаимодействие двух узлов

В сущности, протокол и интерфейс выражают одно и то же понятие, но традиционно в сетях за ними закреплены разные области действия: протоколы определяют правила взаимодействия модулей одного уровня в разных узлах, а интерфейсы – модулей соседних уровней в одном узле.

Средства каждого уровня должны отрабатывать, во-первых, собственный протокол, а во-вторых, интерфейсы с соседними уровнями.

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

Коммуникационные протоколы могут быть реализованы как программно, так и аппаратно. Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней – как правило, чисто программными средствами.

Программный модуль, реализующий некоторый протокол, часто для краткости также называют протоколом. При этом соотношение между протоколом как формально определенной процедурой и протоколом – программным модулем, реализующим эту процедуру, – аналогично соотношению между алгоритмом решения некоторой задачи и программой, решающей эту задачу.

Понятно, что один и тот же алгоритм может быть запрограммирован с разной степенью эффективности. Точно так же и протокол может иметь несколько программных реализаций. Именно поэтому при сравнении протоколов следует учитывать не только логику их работы, но и качество программных решений. Более того, на эффективность взаимодействия устройств в сети влияет качество всей совокупности протоколов, составляющих стек, в частности, то, насколько рационально распределены функции между протоколами разных уровней и насколько хорошо определены интерфейсы между ними.

Протоколы реализуются не только компьютерами, но и другими сетевыми устройствами – концентраторами, мостами, коммутаторами, маршрутизаторами и т.д. Действительно, в общем случае связь компьютеров в сети осуществляется не напрямую, а через различные коммуникационные устройства. В зависимости от типа устройства в нем должны быть встроенные средства, реализующие тот или иной набор протоколов.

6. МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ ОТКРЫТЫХ СИСТЕМ - OSI

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

6.1. Общая характеристика модели OSI

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

В модели OSI (рис. 6.1) средства взаимодействия делятся на **семь уровней**: прикладной, представления, сеансовый, транспортный, сетевой, канальный и физический. Каждый уровень имеет дело с совершенно определенным аспектом взаимодействия сетевых устройств.

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

Итак, пусть приложение узла А хочет взаимодействовать с приложением узла В. Для этого приложение А обращается с запросом к прикладному уровню, например к файловой службе. На основании этого запроса программное обеспечение прикладного уровня формирует сообщение стандартного формата. После формирования сообщения прикладной уровень направляет его вниз по стеку уровню представления. Протокол уровня представления на основании информации, полученной из заголовка сообщения прикладного уровня,

выполняет требуемые действия и добавляет к сообщению собственную служебную информацию - заголовок уровня представления, в котором содержатся указания для протокола уровня представления машины-адресата. Полученное в результате сообщение передается вниз сеансовому уровню, который в свою очередь добавляет свой заголовок и т. д. Наконец, сообщение достигает нижнего, физического уровня, который, собственно, и передает его по линиям связи машине-адресату. К этому моменту сообщение «обрастает» заголовками всех уровней (рис. 6.1).

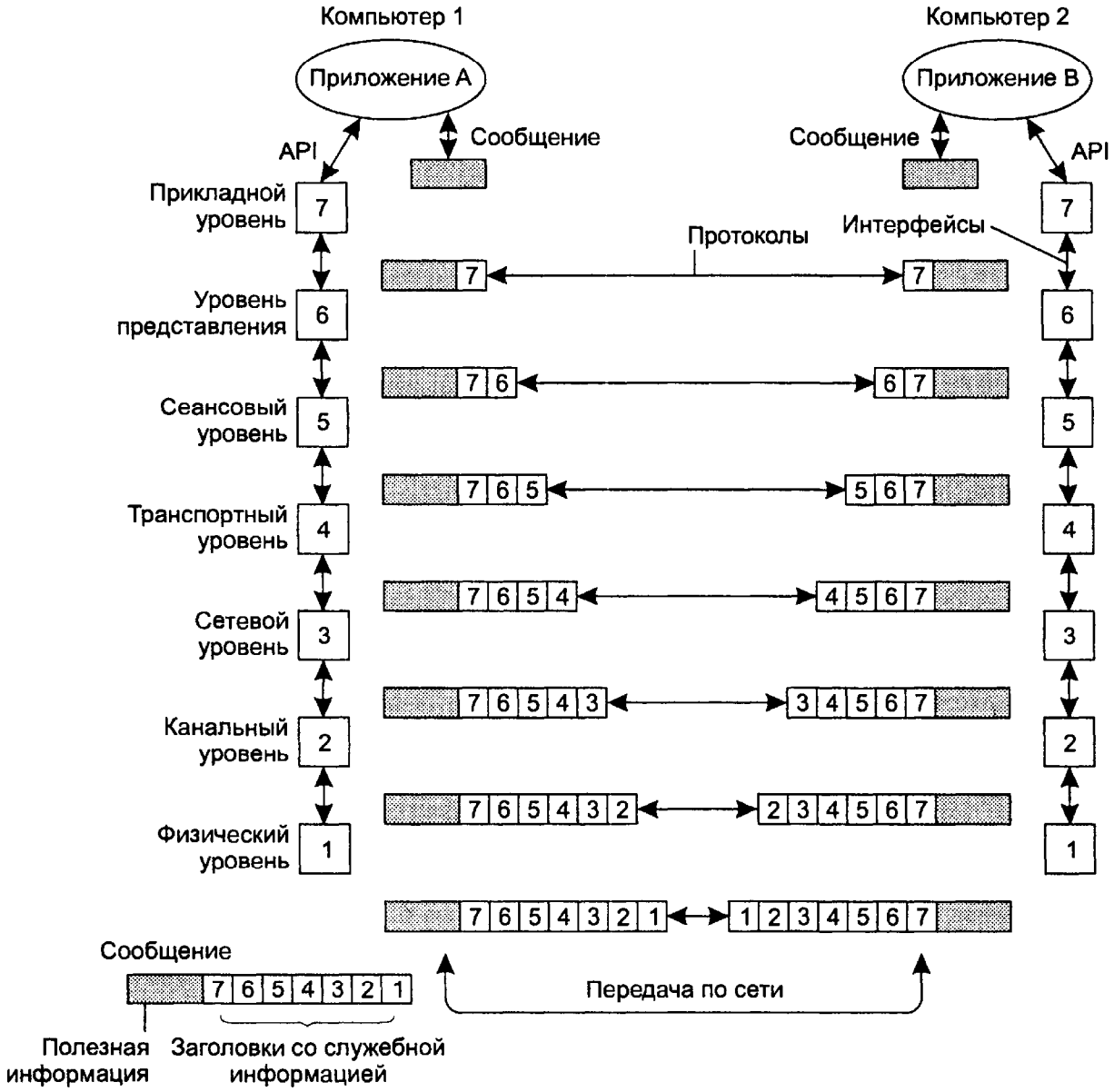


Рисунок 6.1. Модель взаимодействия открытых систем ISO/OSI

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

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

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

6.2. Уровни модели OSI

Физический уровень

Физический уровень (physical layer) имеет дело с передачей потока битов по физическим каналам связи, таким как коаксиальный кабель, витая пара, оптоволоконный кабель или цифровой территориальный канал.

К этому уровню имеют отношение характеристики физических сред передачи данных, такие как полоса пропускания, помехозащищенность, волновое сопротивление и другие. На этом же уровне определяются характеристики электрических сигналов, передающих дискретную информацию, такую как крутизна фронтов импульсов, уровни напряжения или тока передаваемого сигнала, тип кодирования, скорость передачи сигналов. Кроме того, здесь стандартизируются типы разъемов и назначение каждого контакта.

Функции физического уровня:

- передача битов по физическим каналам;
- преобразование данных в передаваемый сигнал, соответствующий физической среде;

• кодирование информации;

• синхронизация;

• модуляция.

Физический уровень реализуется аппаратно. Функции физического уровня реализуются на всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом (COM-порт).

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

Канальный уровень

Канальный уровень (data link layer) является первым уровнем (если идти снизу вверх), который работает в режиме коммутации пакетов. На этом уровне блок данных обычно носит название **кадр** (frame).

Функции средств канального уровня определяются по-разному для локальных и глобальных сетей.

В глобальных сетях канальный уровень должен обеспечивать доставку кадра только между двумя ***соседними*** узлами, соединенными индивидуальной линией связи.

В локальных сетях канальный уровень должен обеспечивать доставку кадра между ***любыми*** узлами сети. При этом предполагается, что сеть имеет типовую топологию, например общую шину, кольцо, звезду или дерево (иерархическую звезду).

Функции канального уровня:

- физическая адресация узлов сети;
- формирование кадров данных соответствующего формата;
- реализация соответствующего метода доступа к общей среде передачи;
- обнаружение и коррекция ошибок.

Адреса, с которыми работает протокол канального уровня, используются для доставки кадров только в пределах этой сети, а для перемещения пакетов между сетями применяются уже адреса следующего, сетевого, уровня.

На канальном уровне байты собираются в **кадры** (frame). Каждый **кадр** – элементарная единица, которая может быть доставлена по назначению. Кадр должен содержать следующие элементы: признак начала кадра; признак конца кадра; адрес источника; адрес получателя; тип кадра; поле данных; контрольная сумма кадра.

Если в сети используется разделяемая среда, то прежде чем физический уровень начнет передавать данные, канальный уровень должен ***проверить доступность общей разделяемой среды.*** Существуют следующие методы доступа к общей среде передачи:

- случайный метод доступа (Ethernet);
- маркерный метод доступа (Token Ring, FDDI);
- арбитражный метод доступа (100VG-AnyLAN);
- ведущий-ведомый (master-slave).

Одной из задач канального уровня является ***обнаружение и коррекция ошибок.*** Для этого канальный уровень фиксирует границы кадра, помещая специальную последовательность битов в его начало и конец, а затем добавляет к кадру контрольную сумму, которая называется также **контрольной последовательностью кадра** (Frame Check Sequence, FCS). Контрольная сумма вычисляется по некоторому алгоритму как функция от всех байтов кадра. По значению FCS узел назначения сможет определить, были ли искажены данные кадра в процессе передачи по сети.

В компьютерах функции канального уровня реализуются совместными усилиями сетевых адаптеров и их драйверов.

Сетевой уровень

Сетевой уровень (network layer) служит для образования единой транспортной системы, объединяющей несколько сетей, причем эти сети могут

использовать различные принципы передачи сообщений между конечными узлами и обладать произвольной структурой связей.

Функции сетевого уровня:

- объединение в общую сеть локальных и глобальных сетей, а также сетей с различными базовыми технологиями;
- логическая (сетевая) адресация узлов сети (IP-адрес);
- продвижение пакетов в составной сети;
- выбор оптимального маршрута доставки пакетов;

Функции сетевого уровня реализуются:

- группой протоколов;
- специальными устройствами — маршрутизаторами.

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

На сетевом уровне определяется два вида протоколов. Первый вид – ***сетевые протоколы*** (routed protocols) – реализуют продвижение пакетов через сеть. Именно эти протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня (IP, IPX). Однако к сетевому уровню относят и другой вид протоколов, называемых протоколами обмена маршрутной информацией или просто ***протоколами маршрутизации*** (routing protocols). С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений (RIP, OSPF, IGRP).

На сетевом уровне работают протоколы еще одного типа, которые отвечают за отображение адреса узла, используемого на сетевом уровне, в локальный адрес сети. Такие протоколы часто называют ***протоколами разрешения адресов*** – Address Resolution Protocol, ARP. Иногда их относят не к сетевому уровню, а к канальному, хотя тонкости классификации не изменяют сути.

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

Транспортный уровень

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

Транспортный уровень (transport layer) обеспечивает приложениям или верхним уровням стека - прикладному, представления и сеансовому - передачу данных с той степенью надежности, которая им требуется.

Функции транспортного уровня:

- обеспечение надежности передачи пакетов между любыми двумя узлами составной сети;
- разбивка сообщения сеансового уровня на пакеты, их нумерация;
- буферизация принимаемых пакетов;
- упорядочивание прибывающих пакетов;
- адресация прикладных процессов;
- управление потоком.

Транспортный уровень, как и канальный, предоставляет функции, отвечающие за целостность и корректность передаваемых данных. Но в отличие от канального уровня, полномочия транспортного уровня распространяются за пределы текущего сегмента локальной сети.

Модель OSI определяет пять **классов транспортного сервиса** от низшего класса 0 до высшего класса 4. Эти виды сервиса отличаются качеством предоставляемых услуг: срочностью, возможностью восстановления прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, а главное — способностью к обнаружению и исправлению ошибок передачи, таких как искажение, потеря и дублирование пакетов.

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

На транспортном уровне существуют два типа протоколов: с установлением логического соединения (TCP) и без установления логического соединения (UDP).

Современные сетевые ОС являются многозадачными, что позволяет пользователю выполнять несколько сетевых программ одновременно. Чтобы разделить данные, приходящие различным приложениям, работающим на одном компьютере в протоколах TCP и UDP используют понятие «порта». Протоколы транспортного уровня присваивают каждому приложению индивидуальный номер порта.

Сеансовый уровень

Сеансовый уровень (session layer) обеспечивает управление взаимодействием сторон: фиксирует, какая из сторон является активной в настоящий момент, и предоставляет средства синхронизации сеанса.

Сеансовый уровень – управление диалогом объектов прикладного уровня:

- установление режима обмена сообщениями (дуплексный или полудуплексный);

- синхронизация обмена сообщениями;

- организация "контрольных точек" диалога.

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

Уровень представления

Уровень представления (presentation layer), как явствует из его названия, обеспечивает представление передаваемой по сети информации, не меняя при этом ее содержания. За счет уровня представления информация, передаваемая прикладным уровнем одной системы, всегда понятна прикладному уровню другой системы. С помощью средств данного уровня протоколы прикладных уровней могут преодолеть синтаксические различия в представлении данных или же различия в кодах символов, например кодов ASCII и EBCDIC. На этом уровне могут выполняться шифрование и дешифрирование данных, благодаря которым секретность обмена данными обеспечивается сразу для всех прикладных служб.

Уровень представления – согласовывает представление (синтаксис) данных при взаимодействии двух прикладных процессов:

- преобразование данных из внешнего формата во внутренний;

- шифрование и расшифровка данных.

- сжатие данных;

- трансляция протоколов – передача данных между разными ОС.

Прикладной уровень

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

Прикладной уровень – набор всех сетевых сервисов, которые предоставляет система конечному пользователю:

- идентификация, проверка прав доступа;

- принт- и файл-сервис, почта, удаленный доступ и.т.д...

Существует очень большое разнообразие протоколов и соответствующих служб прикладного уровня. Приведем в качестве примера несколько наиболее распространенных реализаций сетевых файловых служб: NFS и FTP в стеке TCP/IP, SMB в Microsoft Windows, NCP в операционной системе Novell NetWare.

На рис. 6.2 показано соответствие функций различных коммуникационных устройств уровням модели OSI.

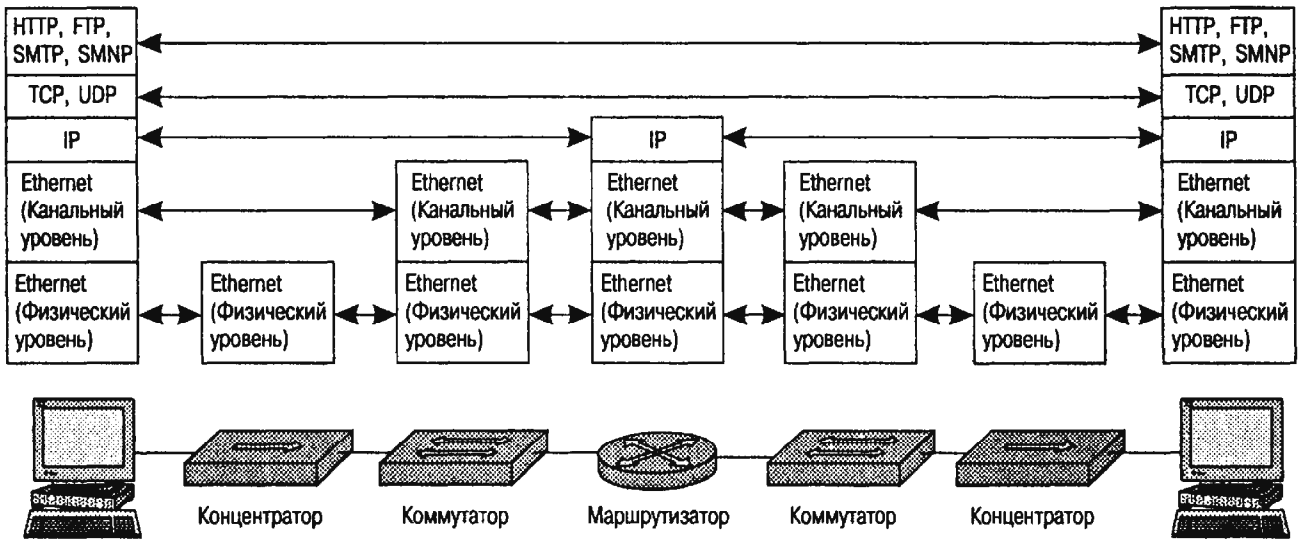


Рисунок 6.2. Соответствие функций различных устройств сети уровням модели OSI

На рис. 6.2 показаны основные элементы компьютерной сети: конечные узлы — компьютеры и промежуточные узлы — коммутаторы и маршрутизаторы (для примера выбраны протоколы стека TCP/IP, как наиболее распространенного).

Из рисунка видно, что полный стек протоколов реализован только на конечных узлах, а промежуточные узлы поддерживают протоколы всех трех

нижних уровней. Это объясняется тем, что коммуникационным устройствам для продвижения пакетов достаточно функциональности нижних трех уровней. Более того, коммуникационное устройство может поддерживать только протоколы двух нижних уровней или даже одного физического уровня — это зависит от типа устройства.

Концентратор — это устройство, которое работает с потоком битов и поэтому ограничивается поддержкой протокола физического уровня.

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

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

Компьютеры, на которых работают сетевые приложения, должны поддерживать протоколы всех уровней. Протоколы прикладного уровня, пользуясь сервисами протоколов уровня представления и сеансового уровня, предоставляют приложениям набор сетевых услуг в виде сетевого интерфейса API. Протокол транспортного уровня также работает на всех конечных узлах. При передаче данных через сеть два модуля транспортного протокола, работающие на узле-отправителе и узле-получателе, взаимодействуют друг с другом для поддержания транспортного сервиса нужного качества. Коммуникационные устройства сети переносят сообщения транспортного протокола прозрачным образом, не вникая в их содержание.

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

7. СТРУКТУРА СТАНДАРТОВ IEEE

В 1980 году в институте IEEE был организован "Комитет 802 по стандартизации локальных сетей", в результате работы которого было принято семейство стандартов IEEE 802.x, которые содержат рекомендации для проектирования нижних уровней локальных сетей.

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

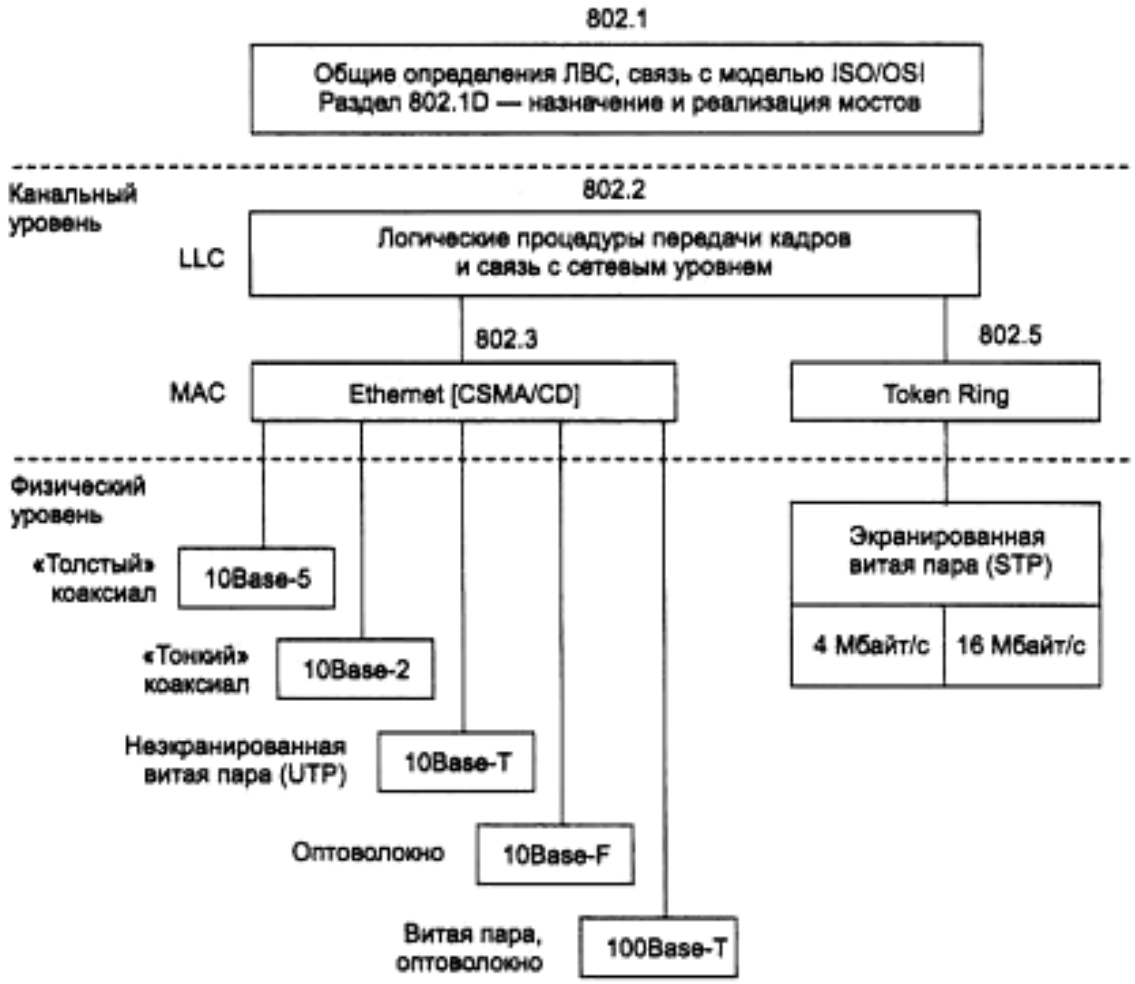


Рисунок 7.1. Структура стандартов IEEE 802.X

Специфика локальных сетей нашла также свое отражение в разделении канального уровня на два подуровня:

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

MAC-уровень появился из-за существования в локальных сетях разделяемой среды передачи данных. Именно этот уровень обеспечивает корректное совместное использование общей среды, предоставляя ее в соответствии с определенным алгоритмом в распоряжение той или иной станции сети.

Уровень LLC отвечает за достоверную передачу кадров данных между узлами, а также реализует функции интерфейса с прилегающим к нему сетевым уровнем. Для уровня LLC также существует несколько вариантов протоколов, отличающихся наличием или отсутствием на этом уровне процедур восстановления кадров в случае их потери или искажения, то есть отличающихся качеством транспортных услуг этого уровня.

Часть стандартов группы 802 описывает отдельные технологии, а часть содержит стандарты, общие для разных технологий.

Подгруппа 802.1 содержит общие определения локальных сетей, связь модели IEEE 802 с моделью OSI, правила взаимодействия различных технологий. К ней относятся:

- 802.1d – логика работы моста/коммутатора; алгоритм покрывающего дерева.
- 802.1h – логика работы транслирующего моста (связывающего сети разных технологий).
- 802.1p – дополнения к логике мостов для работы с трафиком разных приоритетов и выполнения динамической фильтрации группового вещания.
- 802.1q – построение виртуальных локальных сетей (Virtual LAN, VLAN) с помощью мостов/коммутаторов.

Стандарт 802.2 описывает работу подуровня LLC. Стандарты на методы управления логическим соединением.

Подгруппа стандартов 802.3 описывает работу подуровня MAC и физического уровня с методом доступа CSMA/CD. Собственно стандарт 802.3 определяет технологию Ethernet (10 Мбит/с), 802.3u – Fast Ethernet (100 Мбит/с), 802.3z и 802.3ab – Gigabit Ethernet (1 Гбит/с). Стандарт 802.3x определяет правила управления потоком для дуплексного режима.

Стандарт 802.4 описывает работу подуровня MAC и физического уровня технологий типа маркерная шина (Token bus network), прототип - ArcNet протокол MAP (Manufacturing Automation Protocol) для связи устройств промышленной автоматики).

Стандарт 802.5 описывает работу подуровня MAC и физического уровня технологий типа маркерного кольца (Token Ring).

Стандарт 802.6 описывает городские сети (Metropolitan Area Network, MAN).

Стандарт 802.7 описывает принципы широкополосной передачи. Передача разных сигналов по одному широкополосному кабелю методом частотного уплотнения каналов.

Стандарт 802.8 описывает принципы построения сетей на основе волоконно-оптических технологий.

Стандарт 802.9 содержит совместимые с ISDN спецификации совместной передачи голоса и данных.

Стандарт 802.10 описывает принципы сетевой безопасности.

Стандарт 802.11 описывает беспроводные технологии передачи данных (Wi-Fi).

Стандарт 802.12 определяет технологию передачи с методом приоритетного доступа по требованию. Технология 100VG-AnyLAN.

Стандарт 802.15 определяет стандарты на построение персональных сетей (802.15.1 – Bluetooth 1.1)

Стандарт 802.16 стандарт широкополосной беспроводной связи (Wi-Max).

8. ПРОТОКОЛ LLC

Стандарт 802.2 описывает работу подуровня LLC – логические процедуры передачи кадров и связь с сетевым уровнем. В основу протокола LLC положен протокол HDLC (High-level Data Link Control Procedure), широко использующийся в территориальных сетях.

8.1. Три типа процедур уровня LLC

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

- LLC1 - сервис без установления соединения и без подтверждения;
- LLC2 - сервис с установлением соединения и подтверждением;
- LLC3 - сервис без установления соединения, но с подтверждением.

Этот набор процедур является общим для всех методов доступа к среде, определенных стандартами 802.3-802.12.

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

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

В некоторых случаях (например, при использовании сетей в системах реального времени, управляющих промышленными объектами), когда временные издержки установления логического соединения перед отправкой данных

неприемлемы, а подтверждение корректности приема переданных данных необходимо, базовый сервис без установления соединения и без подтверждения не подходит. Для таких случаев предусмотрен дополнительный сервис, называемый сервисом *без установления соединения, но с подтверждением LLC3*.

8.2. Структура кадров LLC

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

- *Информационные кадры* предназначены для передачи информации в процедурах с установлением логического соединения и должны обязательно содержать поле информации. В процессе передачи информационных блоков осуществляется их нумерация в режиме скользящего окна.
- *Управляющие кадры* предназначены для передачи команд и ответов в процедурах с установлением логического соединения, в том числе запросов на повторную передачу искаженных информационных блоков.
- *Ненумерованные кадры* предназначены для передачи ненумерованных команд и ответов, выполняющих в процедурах без установления логического соединения передачу информации, идентификацию и тестирование LLC-уровня, а в процедурах с установлением логического соединения - установление и разъединение логического соединения, а также информирование об ошибках.

Все типы кадров уровня LLC имеют единый формат (рис. 8.1). Они содержат четыре поля:

- адрес точки входа сервиса назначения (Destination Service Access Point, DSAP),
- адрес точки входа сервиса источника (Source Service Access Point, SSAP),
- управляющее поле (Control)
- поле данных (Data)

Флаг (01111110)	Адрес точки входа сервиса назначения **DSAP**	Адрес точки входа сервиса источника **SSAP**	Управляющее поле **Control**	Данные **Data**	Флаг (01111110)

Рисунок 8.1. Структура LLC-кадра стандарта 802.2

Кадр LLC обрамляется двумя однобайтовыми полями "Флаг", имеющими значение 01111110. Флаги используются на MAC-уровне для определения границ блока.

Поле данных кадра LLC предназначено для передачи по сети пакетов протоколов верхних уровней - IP, IPX, AppleTalk, DECnet. Поле данных может отсутствовать в управляющих кадрах и некоторых ненумерованных кадрах.

Поле управления (один байт) используется для обозначения типа кадра данных - информационный, управляющий или ненумерованный. Кроме этого, в этом поле указываются порядковые номера отправленных и успешно принятых кадров, если подуровень LLC работает по процедуре LLC2 с установлением соединения.

Поля DSAP и SSAP позволяют указать, какой сервис верхнего уровня пересылает данные с помощью этого кадра. Программному обеспечению узлов сети при получении кадров канального уровня необходимо распознать, какой протокол вложил свой пакет в поле данных поступившего кадра, для того, чтобы передать извлеченный из кадра пакет нужному протоколу для последующей обработки.

DSAP - это нечто вроде идентификационного номера процесса высшего уровня, который должен принять потом данные. Другими словами, это адрес протокола верхнего уровня, который принимает данные. То есть, когда кадр поступит к получателю, его уровень LLC обработает этот кадр, выполнит, что от него требовалось, а потом процесс высшего уровня, адрес которого указан в поле DSAP, примет оставшиеся данные.

SSAP - это адрес верхнего уровня, который передает данные в протокол LLC.

Таким образом, адреса DSAP и SSAP позволяют указать, какая служба верхнего уровня пересылает данные с помощью этого кадра и какой службе верхнего уровня эти данные предназначены.

Адреса DSAP и SSAP занимают по 1 байту.

Каждый кадр LLC относится к одному из трех типов (в зависимости от значения старших битов поля Control):

- ненумерованный (Unnumbered) – поле Control занимает 1 байт, два старших бита имеют значение 11,

- информационный (Information) – поле Control занимает 2 байта, старший бит установлен в 0,

- управляющий (Supervisory) – поле Control занимает 2 байта, два старших бита имеют значение 10.

9. ТЕХНОЛОГИЯ ETHERNET

Ethernet — это самый распространенный на сегодняшний день стандарт локальных сетей.

Когда говорят Ethernet, то под этим обычно понимают любой из вариантов этой технологии. В более узком смысле Ethernet — это сетевой стандарт, основанный на экспериментальной сети Ethernet Network, которую фирма Xerox разработала и реализовала в 1975 году. Метод доступа был опробован еще раньше: во второй половине 60-х годов в радиосети Гавайского университета использовались различные варианты случайного доступа к общей радиосреде, получившие общее название Aloha. В 1980 году фирмы DEC, Intel и Xerox совместно разработали и опубликовали стандарт Ethernet версии II для сети, построенной на основе коаксиального кабеля, который стал последней версией фирменного стандарта Ethernet. Поэтому фирменную версию стандарта Ethernet называют стандартом Ethernet DIX или Ethernet П.

На основе стандарта Ethernet DIX был разработан стандарт IEEE 802.3, который во многом совпадает со своим предшественником, но некоторые различия все же имеются. В то время как в стандарте IEEE 802.3 различаются уровни MAC и LLC, в оригинальном Ethernet оба эти уровня объединены в единый канальный уровень. В Ethernet DIX определяется протокол тестирования конфигурации (Ethernet Configuration Test Protocol), который отсутствует в IEEE 802.3. Несколько отличается и формат кадра, хотя минимальные и максимальные размеры кадров в этих стандартах совпадают. Часто для того, чтобы отличить Ethernet, определенный стандартом IEEE, и фирменный Ethernet DIX, первый называют технологией 802.3, а за фирменным оставляют название Ethernet без дополнительных обозначений.

9.1. Адресация в сетях Ethernet

Для идентификации получателя информации в технологиях Ethernet используются 6-ти байтовые MAC–адреса.

Формат MAC – адреса обеспечивает возможность использования специфических режимов многоадресной адресации в сети Ethernet и, одновременно, исключить возможность появления в пределах одной локальной сети двух станций которые имели бы одинаковый адрес.

Байт №6	Байт №5	Байт №4	Байт №3	Байт №2	Байт №1
Идентификатор производителя оборудования			Индивидуальный идентификатор устройства		

Для написания MAC адреса могут быть использованы различные формы. Наиболее часто используется шестнадцатеричная форма, в которой пары байтов отделяются друг от друга символами «-»:

00-e0-14-00-00-00

В сетях Ethernet и IEEE 802.3 используются три основных режима формирования адреса назначения:

- Unicast – индивидуальный адрес;
- Multicast – групповой адрес;
- Broadcast – широковещательный адрес.

Первый режим адресации (Unicast) используется в том случае, когда станция - источник адресует передаваемый пакет только одному получателю данных.

Признаком использования режима адресации Multicast является наличие 1 в младшем бите старшего байта идентификатора производителя оборудования.

01-00-0C-CC-CC-CC

Станция сети Ethernet и IEEE 802.3 может также использовать режим адресации типа Broadcast. Адрес станции назначения типа Broadcast кодируется специальным значением:

FF-FF-FF-FF-FF-FF

При использовании данного адреса переданный пакет будет принят всеми станциями, которые находятся в данной сети.

9.2. Метод доступа CSMA/CD

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

Протокол CSMA/CD определяет характер взаимодействия рабочих станций в сети с единой общей для всех устройств средой передачи данных. Все станции имеют равноправные условия по передаче данных. Нет определенной последовательности, в соответствии с которой станции могут получать доступ к среде для осуществления передачи. Именно в этом смысле доступ к среде осуществляется случайным образом. Реализация алгоритмов случайного доступа представляется значительно более простой задачей, чем реализация алгоритмов детерминированного доступа. Поскольку в последнем случае требуется или специальный протокол, контролирующий работу всех устройств сети (например протокол обращения маркера, свойственный сетям Token Ring и FDDI), или специальное выделенное устройство - мастер концентратор, который в определенной последовательности предоставлял бы всем остальным станция возможность передавать (сети Arcnet, 100VG AnyLAN).

Однако сеть со случайным доступом имеет один, пожалуй, главный недостаток - это не совсем устойчивая работа сети при большой загруженности, когда может проходить достаточно большое время, прежде чем данной станции удается передать данные. Виной тому коллизии, которые возникают между станциями, начавшими передачу одновременно или почти одновременно. При возникновении коллизии передаваемые данные не доходят до получателей, а передающим станциям приходится повторно возобновлять передачу — методы

кодирования, используемые в Ethernet, не позволяют выделять сигналы каждой станции из общего сигнала

Коллизия — это нормальная ситуация в работе сетей Ethernet. Для возникновения коллизии не обязательно, чтобы несколько станций начали передачу абсолютно одновременно, такая ситуация маловероятна. Гораздо вероятней, что коллизия возникает из-за того, что один узел начинает передачу раньше другого, но до второго узла сигналы первого просто не успевают дойти к тому времени, когда второй узел решает начать передачу своего кадра. То есть коллизии — это следствие распределенного характера сети.

Алгоритм CSMA/CD для передающей станции приведен на рис.9.2.

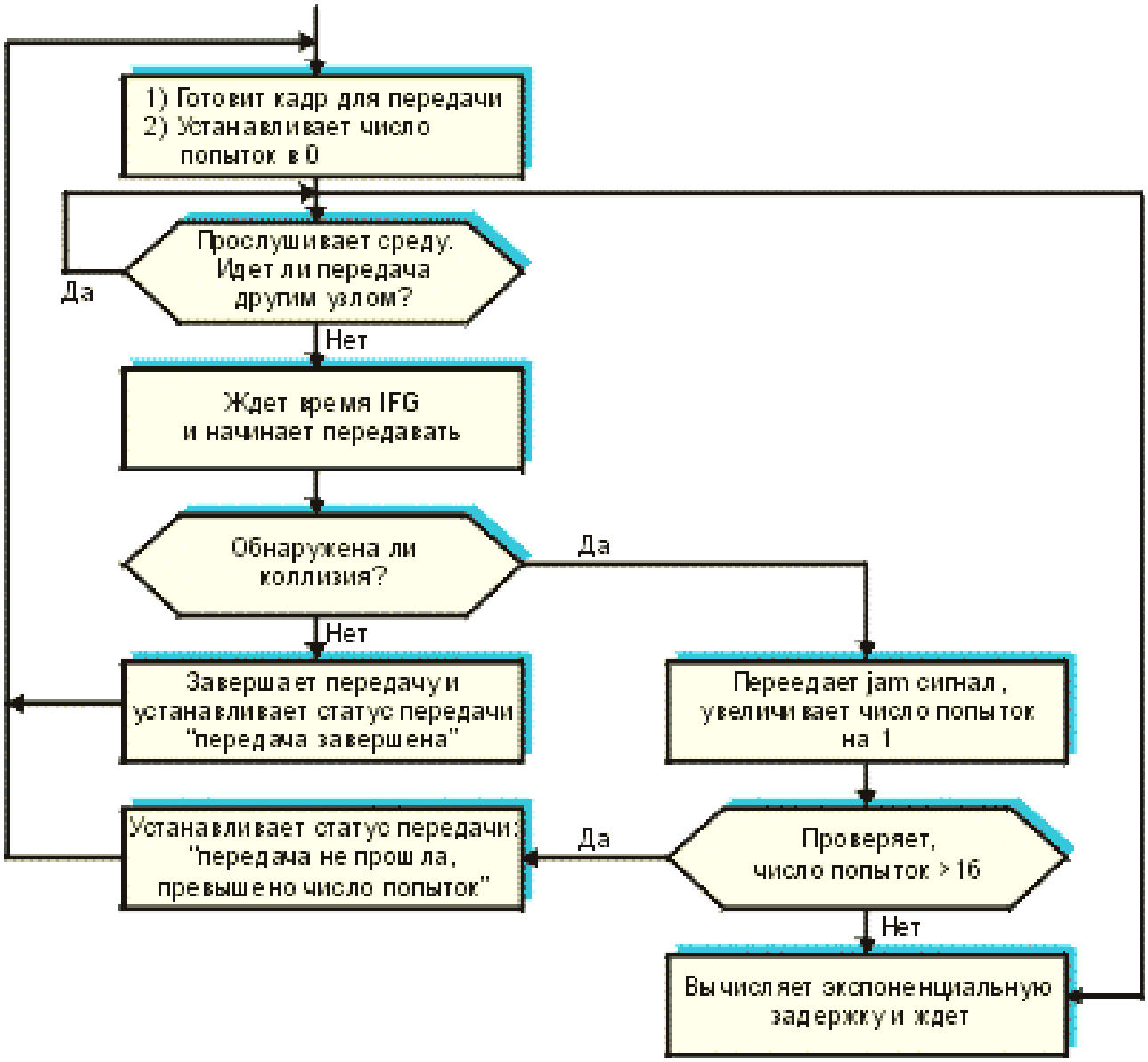


Рисунок 9.2. Структурная схема алгоритма CSMA/CD (уровень MAC): при передаче кадра станцией

Чтобы получить возможность передавать кадр, станция должна убедиться, что разделяемая среда свободна. Это достигается прослушиванием основной гармоники сигнала, которая также называется несущей частотой (carrier-sense, CS). Признаком незанятости среды является отсутствие на ней несущей частоты, которая при манчестерском способе кодирования равна 5-10 МГц, в зависимости от последовательности единиц и нулей, передаваемых в данный момент.

После окончания передачи кадра все узлы сети обязаны выдержать технологическую паузу (Inter Frame Gap) в 9,6 мкс (96 bt). Эта пауза, называемая также межкадровым интервалом, нужна для приведения сетевых адаптеров в исходное состояние, а также для предотвращения монопольного захвата среды одной станцией.

Jam-сигнал (jamming - дословно глушение). Передача jam-сигнала гарантирует, что не один кадр не будет потерян, так как все узлы, которые передавали кадры до возникновения коллизии, приняв jam-сигнал, прервут свои передачи и замолкнут в преддверии новой попытки передать кадры. Jam-сигнал должен быть достаточной длины, чтобы он дошел до самых удаленных станций коллизионного домена, с учетом дополнительной задержки на возможных повторителях.

Четкое распознавание коллизий всеми станциями сети является необходимым условием корректной работы сети Ethernet. Если какая-либо передающая станция не распознает коллизию и решит, что кадр данных ею передан верно, то этот кадр данных будет утерян. Из-за наложения сигналов при коллизии информация кадра исказится, и он будет отбракован принимающей станцией (возможно, из-за несовпадения контрольной суммы). Скорее всего, искаженная информация будет повторно передана каким-либо протоколом верхнего уровня, например транспортным или прикладным, работающим с установлением соединения. Но повторная передача сообщения протоколами

верхних уровней произойдет через значительно более длительный интервал времени (иногда даже через несколько секунд) по сравнению с микросекундными интервалами, которыми оперирует протокол Ethernet. Поэтому если коллизии не будут надежно распознаваться узлами сети Ethernet, то это приведет к заметному снижению полезной пропускной способности данной сети.

Для надежного распознавания коллизий должно выполняться следующее соотношение:

$$\mathbf{T_{min} >= PDV},$$

где T_{min} — время передачи кадра минимальной длины, а PDV — время, за которое сигнал коллизии успевает распространиться до самого дальнего узла сети. Так как в худшем случае сигнал должен пройти дважды между наиболее удаленными друг от друга станциями сети (в одну сторону проходит неискаженный сигнал, а на обратном пути распространяется уже искаженный коллизией сигнал), то именно поэтому это время называется *временем двойного оборота (Path Delay Value, PDV).*

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

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

При возникновении коллизии стация подсчитывает, сколько раз подряд при отправке пакета возникает коллизия. Поскольку повторяющиеся коллизии свидетельствуют о высокой загруженности среды, MAC-узел пытается увеличивать задержку между повторными попытками передачи кадра. Случайная пауза выбирается по следующему алгоритму:

Пауза = L*интервал отсрочки,

где **интервал отсрочки** = 512 битовым интервалам (51,2 мкс);

L представляет собой целое число, выбранное с равной вероятностью из диапазона $[0,2^N]$, где N — номер повторной попытки передачи данного кадра: 1,2,..., 10.

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

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

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

9.3. Форматы кадров технологии Ethernet

В сетях Ethernet на канальном уровне используются кадры 4-х различных форматов. Это связано с длительной историей развития технологии Ethernet, насчитывающей период существования до принятия стандартов IEEE 802, когда подуровень LLC не выделялся из общего протокола и, соответственно, заголовок LLC не применялся.

Различия в форматах кадров могут приводить к несовместимости в работе аппаратуры и сетевого программного обеспечения, рассчитанного на работу только с одним стандартом кадра Ethernet. Однако сегодня практически все сетевые адаптеры, драйверы сетевых адаптеров, мосты/коммутаторы и маршрутизаторы умеют работать со всеми используемыми на практике форматами кадров технологии Ethernet, причем распознавание типа кадра выполняется автоматически.

Ниже приводится описание всех четырех типов кадров Ethernet (здесь под кадром понимается весь набор полей, которые относятся к канальному уровню, то есть поля MAC и LLC уровней). Один и тот же тип кадра может иметь разные названия, поэтому ниже для каждого типа кадра приведено по нескольку наиболее употребительных названий:

- кадр 802.3/LLC (кадр 802.3/802.2 или кадр Novell 802.2);
- кадр Raw 802.3 (или кадр Novell 802.3);
- кадр Ethernet DIX (или кадр Ethernet II);
- кадр Ethernet SNAP.

Форматы всех этих четырех типов кадров Ethernet приведены на рис. 9.3.

Кадр 802.3/LLC

Заголовок кадра 802.3/LLC является результатом объединения полей заголовков кадров, определенных в стандартах IEEE 802.3 и 802.2.

Стандарт 802.3 определяет восемь полей заголовка (рис. 9.3; поле преамбулы и начальный ограничитель кадра на рисунке не показаны).

- *Поле преамбулы (Preamble)* состоит из семи синхронизирующих байт 10101010. При манчестерском кодировании эта комбинация представляется в физической среде периодическим волновым сигналом с частотой 5 МГц.
- *Начальный ограничитель кадра (Start-of-frame-delimiter, SFD)* состоит из одного байта 10101011. Появление этой комбинации бит является указанием на то, что следующий байт — это первый байт заголовка кадра.
- *Адрес назначения (Destination Address, DA)* может быть длиной 2 или 6 байт. На практике всегда используются адреса из 6 байт.
- *Адрес источника (Source Address, SA)* — это 2- или 6-байтовое поле, содержащее адрес узла - отправителя кадра. Первый бит адреса всегда имеет значение 0.
- *Длина (Length, L)* — 2-байтовое поле, которое определяет длину поля данных в кадре.
- *Поле данных (Data)* может содержать от 0 до 1500 байт. Но если длина поля меньше 46 байт, то используется следующее поле — поле заполнения, — чтобы дополнить кадр до минимально допустимого значения в 46 байт.
- *Поле заполнения (Padding)* состоит из такого количества байт заполнителей, которое обеспечивает минимальную длину поля данных в 46 байт. Это обеспечивает корректную работу механизма обнаружения коллизий. Если длина поля данных достаточна, то поле заполнения в кадре не появляется.

- *Поле контрольной суммы (Frame Check Sequence, PCS)* состоит из 4 байт, содержащих контрольную сумму. Это значение вычисляется по алгоритму CRC-32.

Кадр 802.3 является кадром MAC-подуровня, поэтому в соответствии со стандартом 802.2 в его поле данных вкладывается кадр подуровня LLC с удаленными флагами начала и конца кадра. Формат кадра LLC был описан выше. Так как кадр LLC имеет заголовок длиной 3 (в режиме LLC1) или 4 байт (в режиме LLC2), то максимальный размер поля данных уменьшается до 1497 или 1496 байт.

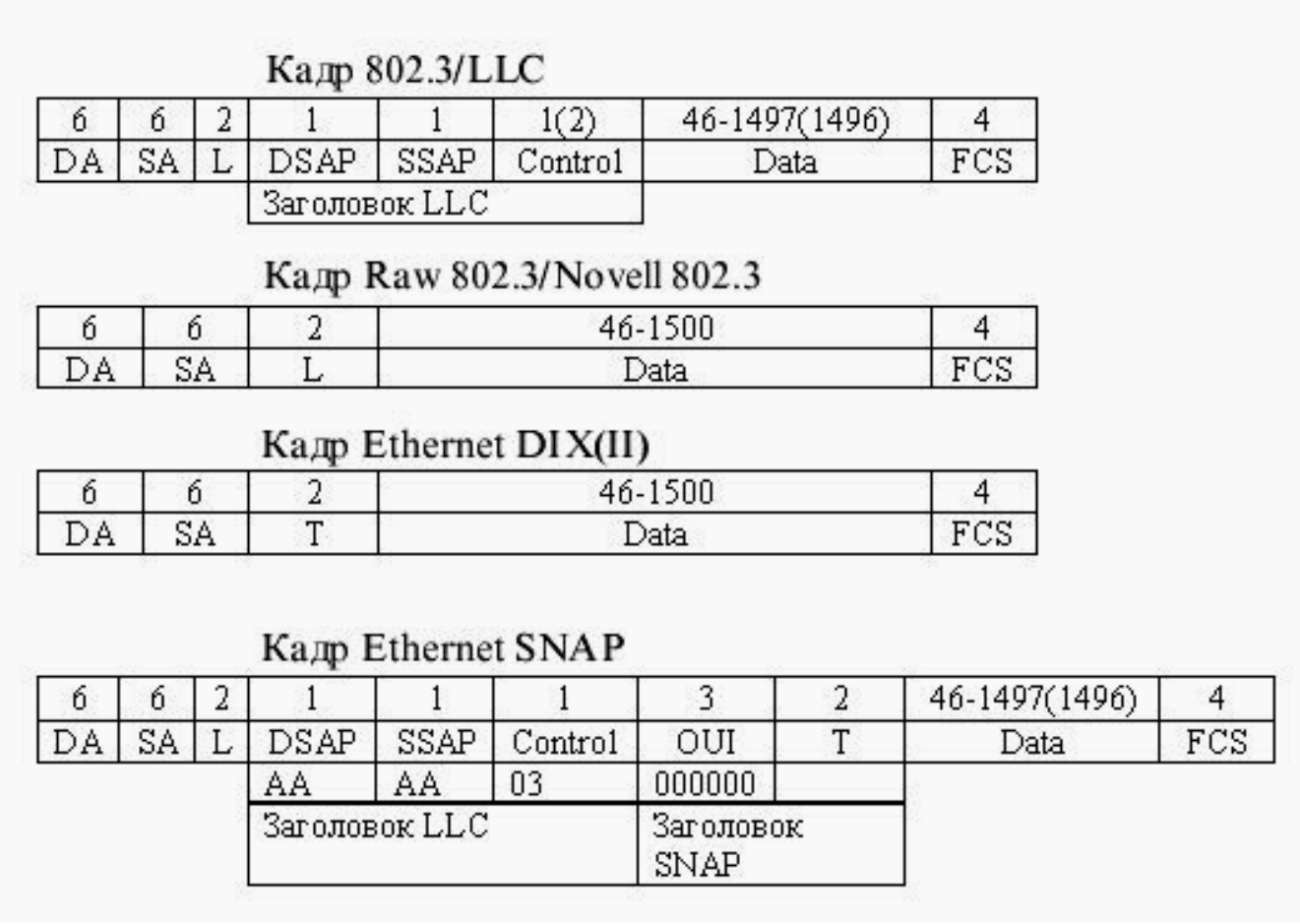


Рисунок 9.3. Форматы кадров Ethernet

Кадр Raw 802.3/Novell 802.3

Кадр Raw 802.3, называемый также кадром Novell 802.3, представлен на рис. 9.3. Из рисунка видно, что это кадр подуровня MAC стандарта 802.3, но без вложенного кадра подуровня LLC. Компания Novell долгое время не

использовала служебные поля кадра LLC в своей операционной системе NetWare из-за отсутствия необходимости идентифицировать тип информации, вложенной в поле данных, — там всегда находился пакет протокола IPX, долгое время бывшего единственным протоколом сетевого уровня в ОС NetWare.

Кадр Ethernet DIX/Ethernet II

Кадр Ethernet DIX, называемый также кадром Ethernet II, имеет структуру (см. рис. 9.3), совпадающую со структурой кадра Raw 802.3. Однако 2-байтовое поле *Длина(L)* кадра Raw 802.3 в кадре ***Ethernet DIX*** используется в качестве поля типа протокола. Это поле, теперь получившее название Type (T) или EtherType, предназначено для тех же целей, что и поля DSAP и SSAP кадра LLC — для указания типа протокола верхнего уровня, вложившего свой пакет в поле данных этого кадра.

Кадр Ethernet SNAP

Для устранения разнобоя в кодировках типов протоколов, сообщения которых вложены в поле данных кадров Ethernet, комитетом 802.2 была проведена работа по дальнейшей стандартизации кадров Ethernet. В результате появился кадр Ethernet SNAP (SNAP — Subnetwork Access Protocol, протокол доступа к подсетям). Кадр Ethernet SNAP (см. рис. 9.3) представляет собой расширение кадра 802.3/LLC за счет введения дополнительного заголовка протокола SNAP, состоящего из двух полей: OUI и Type. Поле Type состоит из 2-х байт и повторяет по формату и назначению поле Type кадра Ethernet II (то есть в нем используются те же значения кодов протоколов). Поле OUI (Organizationally Unique Identifier) определяет идентификатор организации, которая контролирует коды протоколов в поле Type. С помощью заголовка SNAP достигнута совместимость с кодами протоколов в кадрах Ethernet II, а также создана универсальная схема кодирования протоколов. Коды протоколов для технологий 802 контролирует IEEE, которая имеет OUI, равный 000000. Если в будущем потребуются другие коды протоколов для какой-либо новой технологии, для этого достаточно указать другой идентификатор организации, назначающей эти

коды, а старые значения кодов останутся в силе (в сочетании с другим идентификатором OUI).

9.4. Спецификации физической среды Ethernet

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

Физические спецификации технологии Ethernet на сегодняшний день включают следующие среды передачи данных.

- 10Base-5 — коаксиальный кабель диаметром 0,5 дюйма, называемый «толстым» коаксиалом. Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента — 500 метров (без повторителей).
- 10Base-2 — коаксиальный кабель диаметром 0,25 дюйма, называемый «тонким» коаксиалом. Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента — 185 метров (без повторителей).
- 10Base-T — кабель на основе неэкранированной витой пары (Unshielded Twisted Pair, UTP). Образует звездообразную топологию на основе концентратора. Расстояние между концентратором и конечным узлом — не более 100 м.
- 10Base-F — волоконно-оптический кабель. Топология аналогична топологии стандарта 10Base-T. Имеется несколько вариантов этой спецификации — FOIRL (расстояние до 1000 м), 10Base-FL (расстояние до 2000 м), 10Base-FB (расстояние до 2000 м).

Число 10 в указанных выше названиях обозначает битовую скорость передачи данных этих стандартов — 10 Мбит/с, а слово Base — метод передачи

на одной базовой частоте 10 МГц (в отличие от методов, использующих несколько несущих частот, которые называются Broadband — широкополосными). Последний символ в названии стандарта физического уровня обозначает тип кабеля.

Стандарт 10Base-5

Стандарт 10Base-5 в основном соответствует экспериментальной сети Ethernet фирмы Xerox и может считаться классическим Ethernet. Он использует в качестве среды передачи данных коаксиальный кабель с волновым сопротивлением 50 Ом, диаметром центрального медного провода 2,17 мм и внешним диаметром около 10 мм («толстый» Ethernet). Такими характеристиками обладают кабели марок RG-8 и RG-11.

Различные компоненты сети, состоящей из трех сегментов, соединенных повторителями, выполненной на толстом коаксиале, показаны на рис. 9.4.

Кабель используется как моноканал для всех станций. Сегмент кабеля имеет максимальную длину 500 м (без повторителей) и должен иметь на концах согласующие *терминаторы* сопротивлением 50 Ом, поглощающие распространяющиеся по кабелю сигналы и препятствующие возникновению отраженных сигналов. При отсутствии терминаторов («заглушек») в кабеле возникают стоячие волны, так что одни узлы получают мощные сигналы, а другие — настолько слабые, что их прием становится невозможным.

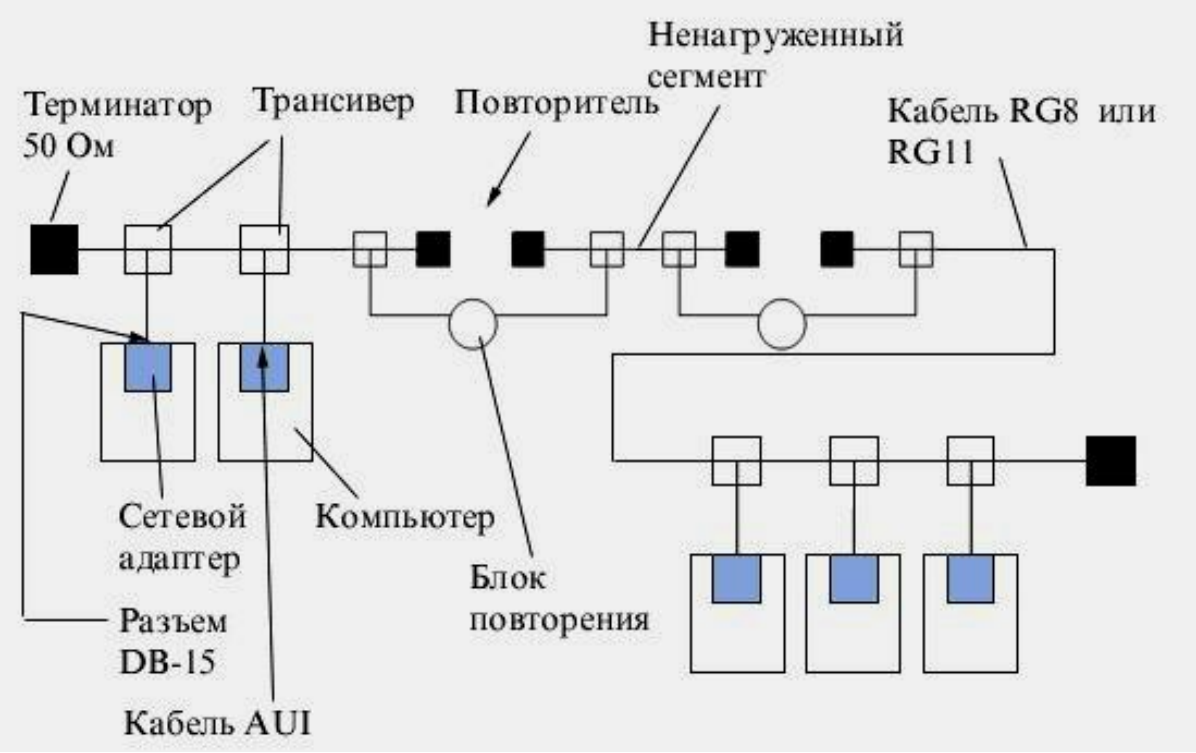


Рисунок 9.4. Компоненты физического уровня сети стандарта 10 Base-5, состоящей из трех сегментов

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

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

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

Стандарт 10Base-5 определяет возможность использования в сети специального устройства — *повторителя (repeator).* Повторитель служит для объединения в одну сеть нескольких сегментов кабеля и увеличения тем самым общей длины сети. Повторитель принимает сигналы из одного сегмента кабеля и побитно синхронно повторяет их в другом сегменте, улучшая форму и мощность импульсов, а также синхронизируя импульсы. Повторитель состоит из двух (или нескольких) трансиверов, которые присоединяются к сегментам кабеля, а также блока повторения со своим тактовым генератором. Для лучшей синхронизации передаваемых бит повторитель задерживает передачу нескольких первых бит преамбулы кадра, за счет чего увеличивается задержка передачи кадра с сегмента на сегмент, а также несколько уменьшается межкадровый интервал.

Правило применения повторителей в сети Ethernet 10Base-5 носит название ***«правило 5-4-3»:*** 5 сегментов, 4 повторителя, 3 нагруженных сегмента. Ограниченное число повторителей объясняется дополнительными задержками распространения сигнала, которые они вносят. Применение повторителей увеличивает время двойного распространения сигнала, которое для надежного распознавания коллизий не должно превышать время передачи кадра минимальной длины, то есть кадра в 72 байт или 576 бит.

Каждый повторитель подключается к сегменту одним своим трансивером, поэтому к нагруженным сегментам можно подключить не более 99 узлов. Максимальное число конечных узлов в сети 10Base-5 таким образом составляет 99х3 = 297 узлов.

К ***достоинствам*** стандарта 10Base-5 относятся:

- хорошая защищенность кабеля от внешних воздействий;

- сравнительно большое расстояние между узлами;
- возможность простого перемещения рабочей станции в пределах длины кабеля AUI.

Недостатками 10Base-5 являются:

- высокая стоимость кабеля;
- сложность его прокладки из-за большой жесткости;
- потребность в специальном инструменте для заделки кабеля;
- останов работы всей сети при повреждении кабеля или плохом соединении;
- необходимость заранее предусмотреть подводку кабеля ко всем возможным местам установки компьютеров.

Стандарт 10Base-2

Стандарт 10Base-2 использует в качестве передающей среды коаксиальный кабель с диаметром центрального медного провода 0,89 мм и внешним диаметром около 5 мм («тонкий» Ethernet). Кабель имеет волновое сопротивление 50 Ом. Такими характеристиками обладают кабели марок RG-58 /U, RG-58A/U, RG-58C/U.

Максимальная длина сегмента без повторителей составляет 185 м, сегмент должен иметь на концах согласующие терминаторы 50 Ом. Тонкий коаксиальный кабель дешевле толстого. Но за дешевизну кабеля приходится расплачиваться качеством — «тонкий» коаксиал обладает худшей помехозащищенностью, худшей механической прочностью и более узкой полосой пропускания.

Станции подключаются к кабелю с помощью высокочастотного BNC T-коннектора, который представляет собой тройник, один отвод которого соединяется с сетевым адаптером, а два других — с двумя концами разрыва кабеля. Максимальное количество станций, подключаемых к одному сегменту, — 30. Минимальное расстояние между станциями —1м. Кабель «тонкого» коаксиала имеет разметку для подключения узлов с шагом в 1 м.

Стандарт 10Base-2 также предусматривает использование повторителей, применение которых также должно соответствовать **«правилу 5-4-3»**. В этом

случае сеть будет иметь максимальную длину в 5x185 = 925 м. Очевидно, что это ограничение является более сильным, чем общее ограничение в 2500 метров.

Типичный состав сети стандарта 10Base-2, состоящей из одного сегмента кабеля, показан на рис. 9.5.

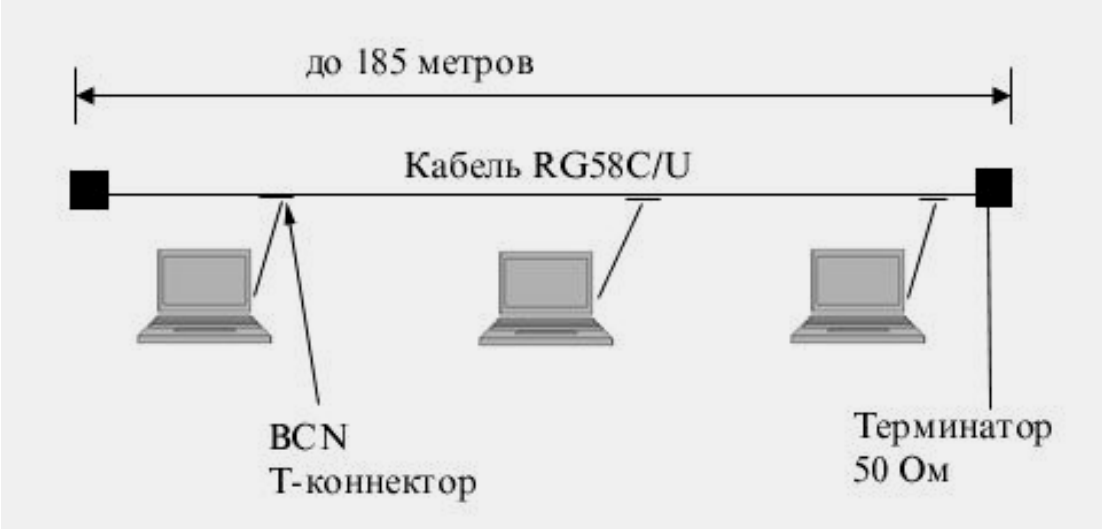


Рисунок 9.5. Сеть стандарта 10Base-2

Общим недостатком стандартов 10Base-5 и 10Base-2 является отсутствие оперативной информации о состоянии моноканала. Повреждение кабеля обнаруживается сразу же (сеть перестает работать), но для поиска отказавшего отрезка кабеля необходим специальный прибор — кабельный тестер.

Стандарт 10Base-T

Стандарт принят в 1991 году, как дополнение к существующему набору стандартов Ethernet, и имеет обозначение 802.31.

Сети 10Base-T используют в качестве среды две *неэкранированные витые пары* (Unshielded Twisted Pair, UTP). Многопарный кабель на основе неэкранированной витой пары категории 3 (категория определяет полосу пропускания кабеля, величину перекрестных наводок NEXT и некоторые другие параметры его качества) телефонные компании уже достаточно давно использовали для подключения телефонных аппаратов внутри зданий.

Конечные узлы соединяются по топологии «точка-точка» со специальным устройством — многопортовым повторителем с помощью двух витых пар. Одна витая пара требуется для передачи данных от станции к повторителю (выход Tx сетевого адаптера), а другая — для передачи данных от повторителя к станции (вход Rx сетевого адаптера). На рис. 9.6 показан пример трехпортового повторителя. Повторитель принимает сигналы от одного из конечных узлов и синхронно передает их на все свои остальные порты, кроме того, с которого поступили сигналы.

Многопортовые повторители в данном случае обычно называются концентраторами (англоязычные термины — hub или concentrator). Концентратор осуществляет функции повторителя сигналов на всех отрезках витых пар, подключенных к его портам, так что образуется единая среда передачи данных — логический моноканал (логическая общая шина). Повторитель обнаруживает коллизию в сегменте в случае одновременной передачи сигналов по нескольким своим Rx-входам и посылает jam-последовательность на все свои Tx-выходы. Стандарт определяет битовую скорость передачи данных 10 Мбит/с и максимальное расстояние отрезка витой пары между двумя непосредственно связанными узлами (станциями и концентраторами) не более 100 м при наличии витой пары качества не ниже категории 3. Это расстояние определяется полосой пропускания витой пары — на длине 100 м она позволяет передавать данные со скоростью 10 Мбит/с при использовании манчестерского кода.

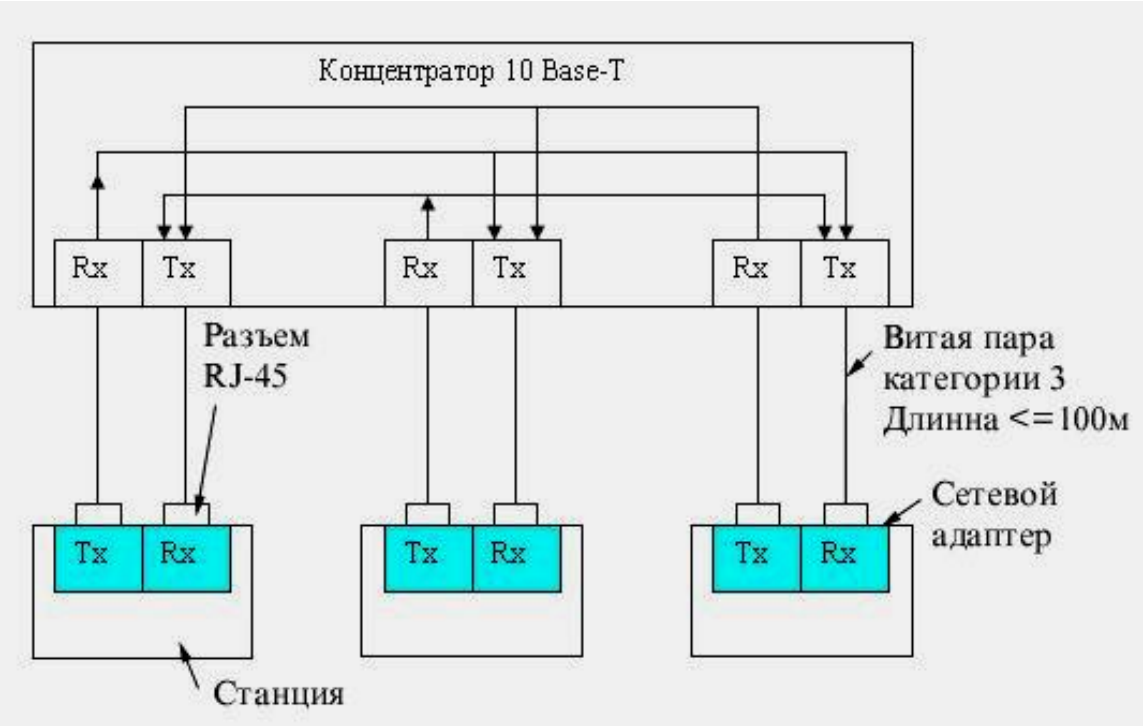


Рисунок 9.6. Сеть стандарта 10Base-T: Tx - передатчик; Rx - приемник

Концентраторы 10Base-T можно соединять друг с другом с помощью тех же портов, которые предназначены для подключения конечных узлов. При этом нужно позаботиться о том, чтобы передатчик и приемник одного порта были соединены соответственно *с приемником и передатчиком другого порта.*

Для обеспечения синхронизации станций при реализации процедур доступа CSMA/CD и надежного распознавания станциями коллизий в стандарте определено максимально число концентраторов между любыми двумя станциями сети, а именно 4. Это правило носит название **«правила 4-х хабов»** и оно заменяет «правило 5-4-3», применяемое к коаксиальным сетям. При создании сети 10Base-T с большим числом станций концентраторы можно соединять друг с другом иерархическим способом, образуя древовидную структуру (рис. 9.7).

Петлевидное соединение концентраторов в стандарте 10Base-T запрещено, так как оно приводит к некорректной работе сети.

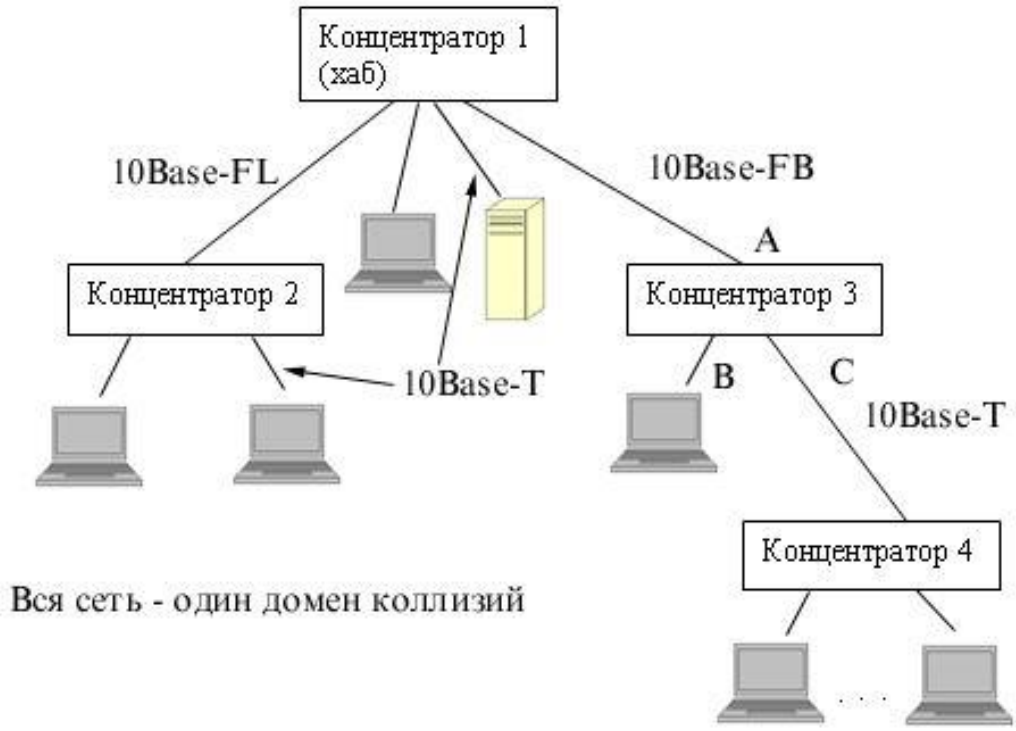


Рисунок 9.7. Иерархическое соединение концентраторов Ethernet

Сети, построенные на основе стандарта 10Base-T, обладают по сравнению с коаксиальными вариантами Ethernet многими преимуществами. Эти преимущества связаны с разделением общего физического кабеля на отдельные кабельные отрезки, подключенные к центральному коммуникационному устройству. И хотя логически эти отрезки по-прежнему образуют общую разделяемую среду, их физическое разделение позволяет контролировать их состояние и отключать в случае обрыва, короткого замыкания или неисправности сетевого адаптера на индивидуальной основе. Это обстоятельство существенно облегчает эксплуатацию больших сетей Ethernet, так как концентратор обычно автоматически выполняет такие функции, уведомляя при этом администратора сети о возникшей проблеме.

В стандарте 10Base-T определена процедура тестирования физической работоспособности двух отрезков витой пары, соединяющих трансивер конечного узла и порт повторителя. Эта процедура называется *тестом связности (link test)*, и она основана на передаче каждые 16 мс специальных импульсов J и K

манчестерского кода между передатчиком и приемником каждой витой пары. Если тест не проходит, то порт блокируется и отключает проблемный узел от сети. Так как коды J и K являются запрещенными при передаче кадров, то тестовые последовательности не влияют на работу алгоритма доступа к среде.

Оптоволоконный Ethernet

В качестве среды передачи данных 10 мегабитный Ethernet использует оптическое волокно. Оптоволоконные стандарты в качестве основного типа кабеля рекомендуют достаточно дешевое многомодовое оптическое волокно, обладающее полосой пропускания 500-800 МГц при длине кабеля 1 км. Функционально сеть Ethernet на оптическом кабеле состоит из тех же элементов, что и сеть стандарта 10Base-T — сетевых адаптеров, многопортового повторителя и отрезков кабеля, соединяющих адаптер с портом повторителя. Как и в случае витой пары, для соединения адаптера с повторителем используются два оптоволокна — одно соединяет выход Tx адаптера со входом Rx повторителя, а другое — вход Rx адаптера с выходом Tx повторителя.

Стандарт FOIRL (Fiber Optic Inter-Repeater Link) представляет собой первый стандарт комитета 802.3 для использования оптоволокна в сетях Ethernet. Он гарантирует длину оптоволоконной связи между повторителями до 1 км при общей длине сети не более 2500 м. Максимальное число повторителей между любыми узлами сети — 4. Максимального диаметра в 2500 м здесь достичь можно, хотя максимальные отрезки кабеля между всеми 4 повторителями, а также между повторителями и конечными узлами недопустимы — иначе получится сеть длиной 5000 м.

Стандарт 10Base-FL представляет собой незначительное улучшение стандарта FOIRL. Увеличена мощность передатчиков, поэтому максимальное расстояние между узлом и концентратором увеличилось до 2000 м. Максимальное число повторителей между узлами осталось равным 4, а максимальная длина сети — 2500 м.

Стандарт 10Base-FB предназначен только для соединения повторителей. Конечные узлы не могут использовать этот стандарт для присоединения к портам концентратора. Между узлами сети можно установить до 5 повторителей 10Base-FB при максимальной длине одного сегмента 2000 м и максимальной длине сети 2740 м.

Повторители, соединенные по стандарту 10Base-FB, при отсутствии кадров для передачи постоянно обмениваются специальными последовательностями сигналов, отличающимися от сигналов кадров данных, для поддержания синхронизации. Поэтому они вносят меньшие задержки при передаче данных из одного сегмента в другой, и это является главной причиной, по которой количество повторителей удалось увеличить до 5. В качестве специальных сигналов используются манчестерские коды J и K в следующей последовательности: J-J-K-K-J-J-... Эта последовательность порождает импульсы частоты 2,5 МГц, которые и поддерживают синхронизацию приемника одного концентратора с передатчиком другого. Поэтому стандарт 10Base-FB имеет также название *синхронный Ethernet.*

В технологии Ethernet, независимо от применяемого стандарта физического уровня, существует понятие домена коллизий.

Домен коллизий (collision domain) — это часть сети Ethernet, все узлы которой распознают коллизию независимо от того, в какой части этой сети коллизия возникла. Сеть Ethernet, построенная на повторителях, всегда образует один домен коллизий. Домен коллизий соответствует одной разделяемой среде. Мосты, коммутаторы и маршрутизаторы делят сеть Ethernet на несколько доменов коллизий.

9.5. Методика расчета конфигурации сети Ethernet

Соблюдение многочисленных ограничений, установленных для различных стандартов физического уровня сетей Ethernet, гарантирует корректную работу сети (естественно, при исправном состоянии всех элементов физического уровня).

Чтобы сеть Ethernet, состоящая из сегментов различной физической природы, работала корректно, необходимо выполнение четырех основных условий:

- количество станций в сети не более 1024;
- максимальная длина каждого физического сегмента не более величины, определенной в соответствующем стандарте физического уровня;
- время двойного оборота сигнала (Path Delay Value, PDV) между двумя самыми удаленными друг от друга станциями сети не более 575 битовых интервала;
- сокращение межкадрового интервала IPG (Path Variability Value, PVV) при прохождении последовательности кадров через все повторители должно быть не больше, чем 49 битовых интервала.

Соблюдение этих требований обеспечивает корректность работы сети даже в случаях, когда нарушаются простые правила конфигурирования, определяющие максимальное количество повторителей и общую длину сети в 2500 м.

Расчет PDV

Для упрощения расчетов обычно используются справочные данные IEEE, содержащие значения задержек распространения сигналов в повторителях, приемопередатчиках и различных физических средах. В табл. 9.1 приведены данные, необходимые для расчета значения PDV для всех физических стандартов сетей Ethernet. Битовый интервал обозначен как bt.

Комитет 802.3 старался максимально упростить выполнение расчетов, поэтому данные, приведенные в таблице, включают сразу несколько этапов прохождения сигнала. Например, задержки, вносимые повторителем, состоят из задержки входного трансивера, задержки блока повторения и задержки выходного трансивера. Тем не менее, в таблице все эти задержки представлены одной величиной, названной базой сегмента. Чтобы не нужно было два раза складывать

задержки, вносимые кабелем, в таблице даются удвоенные величины задержек для каждого типа кабеля.

Таблица 9.1. Данные для расчета значения PDV

Тип сегмента	**База левого сегмента, bt**	**База промежуточного сегмента, bt**	**База правого сегмента, bt**	**Задержка среды на 1 м, bt**	**Максимальная длина сегмента, м**
10Base-5	11,8	46,5	169,5	0,0866	500
10Base-2	11,8	46,5	169,5	0,1026	185
10Base-T	15,3	42,0	165,0	0,113	100
10Base-FB	—	24,0	—	0,1	2000
10Base-FL	12,3	33,5	156,5	0,1	2000
FOIRL	7,8	29,0	152,0	0,1	1000
AUI (> 2 м)	0	0	0	0,1026	2+48

В таблице используются также такие понятия, как левый сегмент, правый сегмент и промежуточный сегмент. Поясним эти термины на примере сети, приведенной на рис. 9.8. Левым сегментом называется сегмент, в котором начинается путь сигнала от выхода передатчика конечного узла. На примере это сегмент 1. Затем сигнал проходит через промежуточные сегменты 2-5 и доходит до приемника наиболее удаленного узла наиболее удаленного сегмента 6, который называется правым. Именно здесь в худшем случае происходит столкновение кадров и возникает коллизия, что и подразумевается в таблице.

С каждым сегментом связана постоянная задержка, названная базой, которая зависит только от типа сегмента и от положения сегмента на пути сигнала (левый, промежуточный или правый). База правого сегмента, в котором возникает коллизия, намного превышает базу левого и промежуточных сегментов.

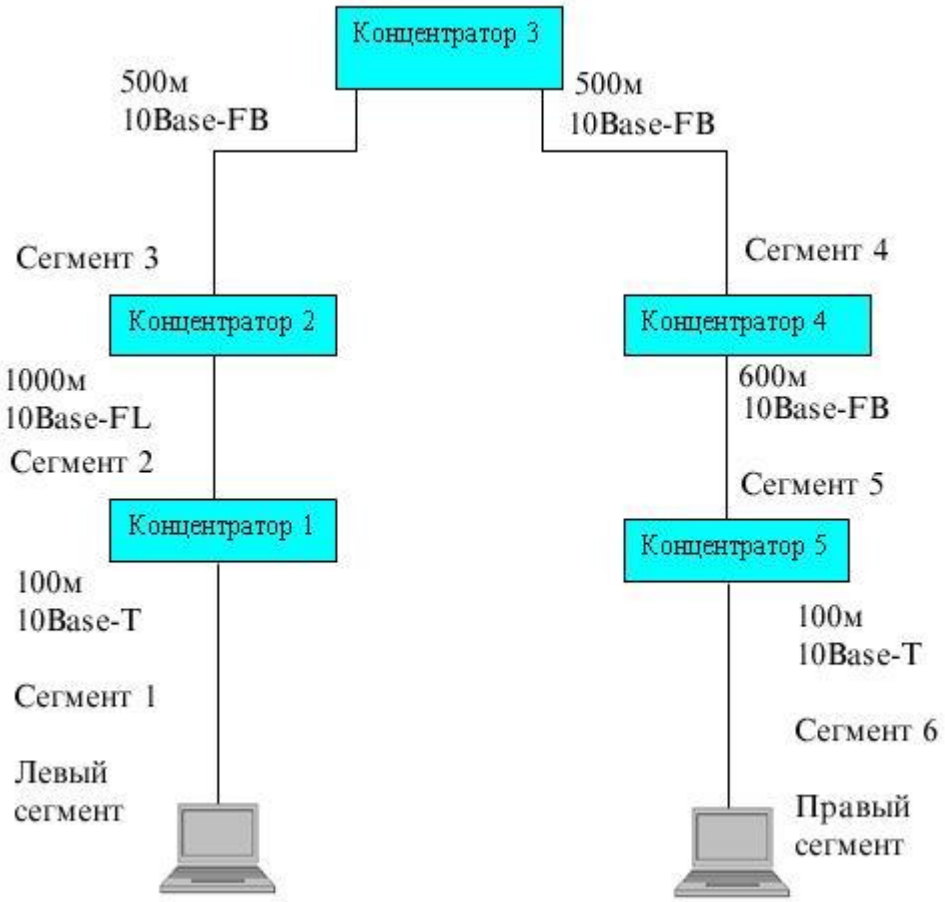


Рисунок 9.8. Пример сети Ethernet, состоящей из сегментов различных физических стандартов

Кроме этого, с каждым сегментом связана задержка распространения сигнала вдоль кабеля сегмента, которая зависит от длины сегмента и вычисляется путем умножения времени распространения сигнала по одному метру кабеля (в битовых интервалах) на длину кабеля в метрах.

Расчет заключается в вычислении задержек, вносимых каждым отрезком кабеля (приведенная в таблице задержка сигнала на 1 м кабеля умножается на длину сегмента), а затем суммировании этих задержек с базами левого, промежуточных и правого сегментов. Общее значение PDV не должно превышать 575.

Так как левый и правый сегменты имеют различные величины базовой задержки, то в случае различных типов сегментов на удаленных краях сети необходимо выполнить расчеты дважды: один раз принять в качестве левого сегмента сегмент одного типа, а во второй — сегмент другого типа. Результатом

можно считать максимальное значение PDV. В нашем примере крайние сегменты сети принадлежат к одному типу — стандарту 10Base-T, поэтому двойной расчет не требуется, но если бы они были, сегментами разного типа, то в первом случае нужно было бы принять в качестве левого сегмент между станцией и концентратором 1, а во втором считать левым сегмент между станцией и концентратором 5.

Приведенная на рисунке сеть в соответствии с правилом 4-х хабов не является корректной — в сети между узлами сегментов 1 и 6 имеется 5 хабов, хотя не все сегменты являются сегментами 10Base-FB. Кроме того, общая длина сети равна 2800 м, что нарушает правило 2500 м. Рассчитаем значение PDV для нашего примера.

Левый сегмент 1: 15,3 (база) + 100 х 0,113 - 26,6.

Промежуточный сегмент 2*:* 33,5 + 1000 х 0,1 = 133,5.

Промежуточный сегмент 3*:* 24 + 500 х 0,1 = 74,0.

Промежуточный сегмент 4*:* 24 + 500 х 0,1 = 74,0.

Промежуточный сегмент 5: 24 + 600 х 0,1 = 84,0.

Правый сегмент 6*:* 165 + 100 х 0,113 = 176,3.

Сумма всех составляющих дает значение PDV, равное 568,4.

Так как значение PDV меньше максимально допустимой величины 575, то эта сеть проходит по критерию времени двойного оборота сигнала несмотря на то, что ее общая длина составляет больше 2500 м, а количество повторителей — больше 4-х.

Расчет PVV

Чтобы признать конфигурацию сети корректной, нужно рассчитать также уменьшение межкадрового интервала повторителями, то есть величину PVV.

Для расчета PVV также можно воспользоваться значениями максимальных величин уменьшения межкадрового интервала при прохождении повторителей различных физических сред, рекомендованными IEEE и приведенными в табл. 9.2.

Таблица 9.2. Сокращение межкадрового интервала повторителями

Тип сегмента	**Передающий сегмент, bt**	**Промежуточный сегмент, bt**
10Base-5 или 10Base-2	16	11
10Base-FB	—	2
10Base-FL	10,5	8
10Base-T	10,5	8

В соответствии с этими данными рассчитаем значение PVV для нашего примера.

Левый сегмент 1: 10Base-T: сокращение в 10,5 bt.

Промежуточный сегмент 2: 10Base-FL: 8.

Промежуточный сегмент 3: 10Base-FB: 2.

Промежуточный сегмент 4: 10Base-FB: 2.

Промежуточный сегмент 5: 10Base-FB: 2.

Сумма этих величин дает значение PVV, равное 24,5, что меньше предельного значения в 49 битовых интервала.

В результате приведенная в примере сеть соответствует стандартам Ethernet по всем параметрам, связанным и с длинами сегментов, и с количеством повторителей.

10. ТЕХНОЛОГИЯ TOKEN RING

10.1. Основные характеристики стандарта Token Ring

Сети стандарта Token Ring, также как и сети Ethernet, используют разделяемую среду передачи данных, которая состоит из отрезков кабеля, соединяющих все станции сети в кольцо. Кольцо рассматривается как общий разделяемый ресурс, и для доступа к нему используется не случайный алгоритм, как в сетях Ethernet, а детерминированный, основанный на передаче станциями права на использование кольца в определенном порядке. Право на использование кольца передается с помощью кадра специального формата, называемого **маркером** или **токеном**.

Стандарт Token Ring был принят комитетом 802.5 в 1985 году. В это же время компания IBM приняла стандарт Token Ring в качестве своей основной сетевой технологии. В настоящее время именно компания IBM является основным законодателем моды технологии Token Ring, производя около 60% сетевых адаптеров этой технологии.

Сети Token Ring работают с двумя битовыми скоростями - 4 Мбит/с и 16 Мбит/с. Первая скорость определена в стандарте 802.5, а вторая является новым стандартом де-факто, появившимся в результате развития технологии Token Ring. Смешение станций, работающих на различных скоростях, в одном кольце не допускается.

Сети Token Ring, работающие со скоростью 16 Мб/с, имеют и некоторые усовершенствования в алгоритме доступа по сравнению со стандартом 4 Мб/с.

10.2. Маркерный метод доступа к разделяемой среде

В сетях с ***маркерным методом доступа*** право на доступ к среде передается циклически от станции к станции по логическому кольцу. Кольцо образуется отрезками кабеля, соединяющими соседние станции. Таким образом, каждая станция связана со своей предшествующей и последующей станцией и может непосредственно обмениваться данными только с ними. Для обеспечения доступа

станций к физической среде по кольцу циркулирует кадр специального формата и назначения - ***маркер (токен)***.

Получив маркер, станция анализирует его, при необходимости модифицирует и при отсутствии у нее данных для передачи обеспечивает его продвижение к следующей станции. Станция, которая имеет данные для передачи, при получении маркера изымает его из кольца, что дает ей право доступа к физической среде и передачи своих данных. Затем эта станция выдает в кольцо кадр данных установленного формата последовательно по битам. Переданные данные проходят по кольцу всегда в одном направлении от одной станции к другой.

При поступлении кадра данных к одной или нескольким станциям, эти станции копируют для себя этот кадр и вставляют в этот кадр подтверждение приема. Станция, выдавшая кадр данных в кольцо, при обратном его получении с подтверждением приема изымает этот кадр из кольца и выдает новый маркер для обеспечения возможности другим станциям сети передавать данные.

На рисунке 10.1 описанный алгоритм доступа к среде иллюстрируется временной диаграммой. Здесь показана передача пакета А в кольце, состоящем из 6 станций, от станции 1 к станции 3.

Время удержания одной станцией маркера ограничивается ***тайм-аутом удержания маркера*** (по умолчанию – 10 мс), после истечения которого станция обязана передать маркер далее по кольцу.

В сетях Token Ring 16 Мбит/с используется также несколько другой алгоритм доступа к кольцу, называемый алгоритмом ***раннего освобождения маркера*** ***(Early Token Release).*** В соответствии с ним станция передает маркер доступа следующей станции сразу же после окончания передачи последнего бита кадра, не дожидаясь возвращения по кольцу этого кадра с битом подтверждения приема. В этом случае пропускная способность кольца используется более эффективно и приближается к 80 % от номинальной.

Для различных видов сообщений передаваемым данным могут назначаться различные ***приоритеты***.

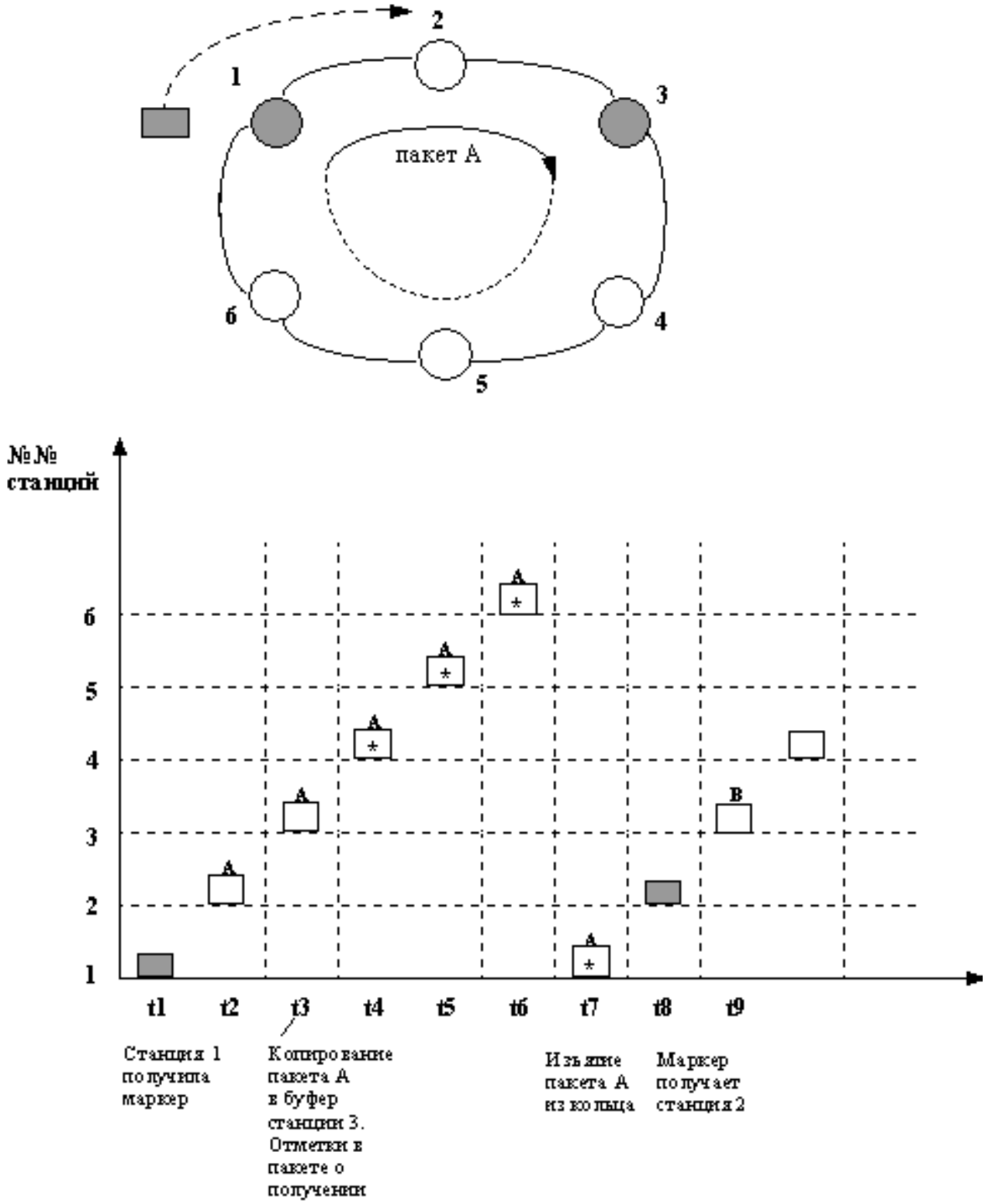


Рисунок. 10.1. Принцип маркерного доступа

Каждая станция имеет механизмы обнаружения и устранения неисправностей сети, возникающих в результате ошибок передачи или переходных явлений (например, при подключении и отключении станции).

Не все станции в кольце равны. Одна из станций обозначается как ***активный монитор***, что означает дополнительную ответственность по управлению кольцом. Активный монитор осуществляет управление тайм-аутом в кольце, порождает новые маркеры (если необходимо), чтобы сохранить рабочее состояние, и генерирует диагностические кадры при определенных обстоятельствах. Активный монитор выбирается, когда кольцо инициализируется, и в этом качестве может выступить любая станция сети. Если монитор отказал по какой-либо причине, существует механизм, с помощью которого другие станции (резервные мониторы) могут договориться, какая из них будет новым активным монитором.

12.3. Форматы кадров Token Ring

В Token Ring существует три различных формата кадров:

- маркер;
- кадр данных;
- прерывающая последовательность.

Маркер

Маркер сети *Token-Ring* представляет собой управляющий кадр, содержащий всего три байта (рис.10.2): байт начального разделителя (SD – Start Delimiter), байт управления доступом (AC – Access Control) и байт конечного разделителя (ED – End Delimiter). Все эти три байта входят также в состав информационного кадра, правда, функции их в маркере и в информационнм кадре несколько различаются.

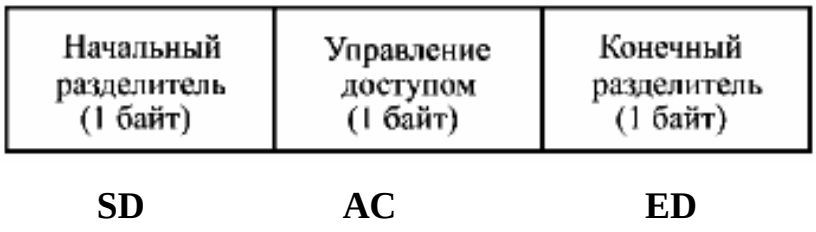


Рисунок 10.2. Формат маркера сети Token-Ring

Начальный разделитель SD содержит четыре нестандартных битовых интервала (рис.10.3). Два из них, обозначающихся J, представляют собой низкий уровень сигнала в течение всего битового интервала. Два других бита, обозначающихся К, представляют собой высокий уровень сигнала в течение всего битового интервала. Понятно, что такие сбои в синхронизации легко выявляются приемником. Биты J и K никогда не могут встречаться среди битов полезной информации.

Рисунок 10.3. Форматы начального (SD) и конечного (ED) разделителей

Конечный разделитель ED также содержит в себе четыре бита специального вида (два бита J и два бита K), а также два единичных бита. Но, кроме того, в него входят и два информационных бита, которые имеют смысл только в составе информационного кадраа:

- Бит I (Intermediate) представляет собой признак промежуточного пакета (1 соответствует первому в цепочке или промежуточному пакету, 0 – последнему в цепочке или единственному пакету).
- Бит E (Error) является признаком обнаруженной ошибки (0 соответствует отсутствию ошибок, 1 – их наличию).

Байт управления доступом (AC – Access Control) разделен на четыре поля (рис.10.4): поле приоритета - три бита **PPP**, бит маркера -**T**, бит монитора - **M** и поле резервирования - три бита **RRR**.

Биты приоритета **PPP** позволяют абоненту присваивать приоритет своим кадрам или маркеру (приоритет может быть от 0 до 7, причем 7 соответствует наивысшему приоритету, а 0 – низшему). Абонент может присоединить к маркеру свой кадр только тогда, когда его собственный приоритет (приоритет его кадров) такой же или выше приоритета маркера.

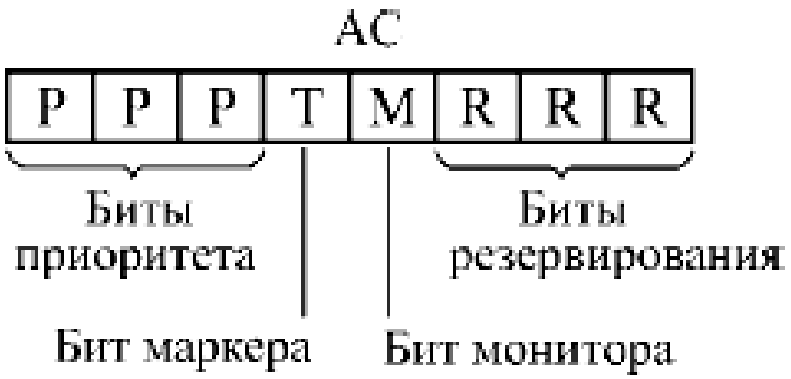


Рисунок 10.4. Формат байта управления доступом

Бит маркера **T** определяет, присоединен ли к маркеру кадр или нет (единица соответствует маркеру без кадра, нуль – маркеру с кадром). Бит монитора **M**, установленный в единицу, говорит о том, что данный маркер передан активным монитором.

Биты резервирования **RRR** позволяют абоненту зарезервировать свое право на дальнейший захват сети, то есть занять очередь на обслуживание.

Формат кадра данных *Token-Ring* представлен на рис.10.5. Помимо начального и конечного разделителей, а также байта управления доступом в этот кадр входят также байт управления пакетом, сетевые адреса приемника и передатчика, данные, контрольная сумма и байт состояния пакета.

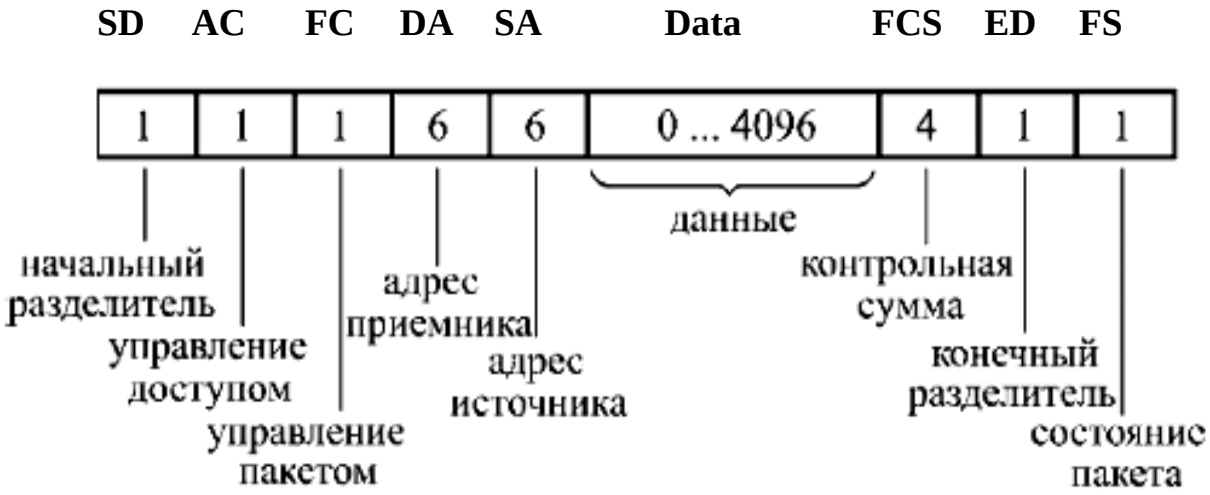


Рисунок 10.5. Формат кадра сети Token-Ring (длина полей дана в байтах)

Назначение полей кадра данных.

- Начальный разделитель (**SD**) является признаком начала кадра, формат – такой же, как и в маркере.
- Поле управления доступом (**AC**) имеет тот же формат, что и в маркере.
- Поле управления кадра (**FC** – Frame Control) определяет тип кадра.

F	**F**	**Z**	**Z**	**Z**	**Z**	**Z**	**Z**

FF – биты типа кадра: 2 бита типа кадра имеют значения 00 для кадров MAC и 01 для кадров LLC.

ZZZZZZ - биты идентификатора управления MAC определяют тип кадра управления кольцом из приведенного ниже списка 6-ти управляющих кадров MAC.

- Шестибайтовые MAC-адреса отправителя и получателя кадра имеют такой же формат, как и в Ethernet.
- Поле данных (**Data**) включает в себя передаваемые данные (в информационном кадре) или информацию для управления обменом (в управляющем кадре).
- Поле контрольной суммы (**FCS** – Frame Check Sequence) представляет собой 32-разрядную циклическую контрольную сумму кадра (CRC).
- Конечный разделитель (**ED**), как и в маркере, указывает на конец кадра. Кроме того, он определяет, является ли данный кадр промежуточным или заключительным в последовательности передаваемых кадров, а также содержит признак ошибочности кадра (см. рис.12.3).
- Поле состояния пакета (**FS** – Frame Status) говорит о том, что происходило с данным кадром: был ли он увиден приемником (то есть, существует ли приемник с заданным адресом) и скопирован в память приемника. По нему отправитель кадра узнает, дошел ли кадр по назначению и без ошибок или его надо передавать заново.

A	**C**	**X**	**X**	**A**	**C**	**X**	**X**

A – бит распознавания адреса; **C** – бит копирования кадра;

На основании значений этих битов, станция-источник кадра различает следующие результаты передачи кадра:

- станция-адресат не существует или не активна (**A=0; C=0**);
- станция-адресат существует, но кадр не скопирован (**A=1; C=0**);
- кадр принят.

Если оба бита установлены (**A=1; C=1**), и установлен бит ошибки (**E=1**), то станция-источник знает, что ошибка возникла после того, как кадр был корректно получен.

Прерывающая последовательность

Состоит из двух байтов, содержащих начальный ограничитель и конечный ограничитель. Прерывающая последовательность может появиться в любом месте потока битов и сигнализирует о том, что текущая передача кадра или маркера отменяется.

Как видно из описания процедур обмена данными, в сети Token Ring на уровнях MAC и LLC применяются процедуры без установления связи, но с подтверждением получения кадров.

10.4. Физическая реализация сетей Token Ring

Сеть ***Token-Ring*** имеет топологию кольцо, хотя внешне она больше напоминает звезду. Это связано с тем, что отдельные абоненты (компьютеры) присоединяются к сети не напрямую, а через специальные концентраторы или многостанционные устройства доступа (MSAU или MAU – Multistation Access Unit). Физически сеть образует звездно-кольцевую топологию (рис.10.6). В действительности же абоненты объединяются все-таки в кольцо, то есть каждый из них передает информацию одному соседнему абоненту, а принимает информацию от другого.

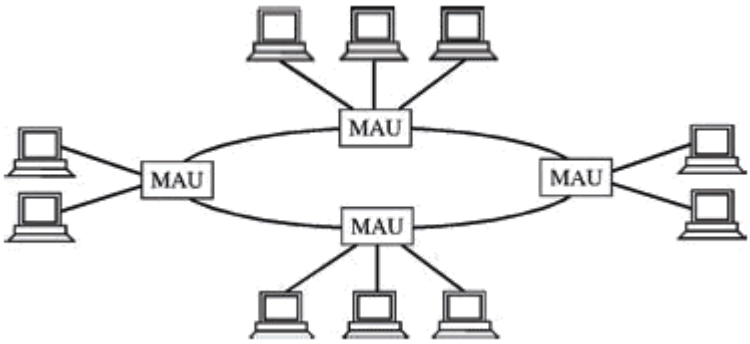


Рисунок 10.6. Звездно-кольцевая топология сети Token-Ring

Концентратор (MAU) при этом позволяет централизовать задание конфигурации, отключение неисправных абонентов, контроль работы сети и т.д. (рис.10.7). Никакой обработки информации он не производит.

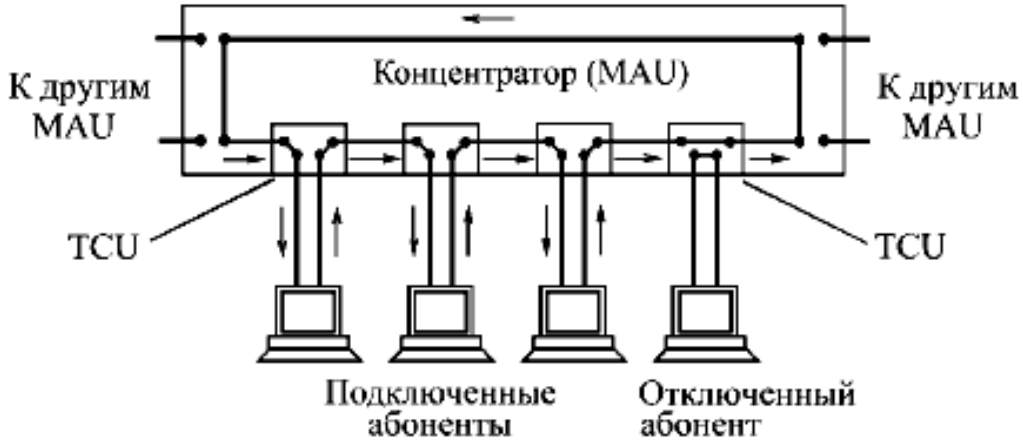


Рисунок 10.7. Соединение абонентов сети Token-Ring в кольцо с помощью концентратора (MAU)

Максимальная длина ответвительного кабеля зависит от типа концентратора, типа кабеля и скорости передачи данных. Обычно для скорости 16 Мб/с максимальная длина кабеля Type 1 может достигать 200 м, а для скорости 4 Мб/с - 600 м. Концентраторы Token Ring делятся на активные и пассивные. Пассивные концентраторы обеспечивают только соединения портов внутри концентратора в кольцо, активные выполняют и функции повторителя, обеспечивая ресинхронизацию сигналов и исправление их амплитуды и формы.

Естественно, что активные концентраторы поддерживают большие расстояния до станции, чем пассивные.

Остальные станции сети соединены в кольцо непосредственными связями. Такие связи называются магистральными (trunk cable). Обычно связи такого рода используются для соединения концентраторов друг с другом для образования общего кольца. Порты концентраторов, предназначенные для такого соединения, называются портами **Ring-In** и **Ring-Out**.

Для предотвращения влияния отказавшей или отключенной станции на работу кольца станции подключаются к магистрали кольца через специальные устройства, называемые устройствами подключения к магистрали (Trunk Coupling Unit, TCU). В функции такого устройства входит образование обходного пути, исключающего заход магистрали в MAC-узел станции при ее отключении или отказе. Обычно для этих целей в TCU используются реле, которые подпитываются постоянным током во время нормальной работы. При пропадании тока подпитки контакты реле переключаются и образуют обходной путь, исключая станцию.

При подключении станции в кольцо через концентратор, устройства TCU встраивают в порты концентратора.

Основные технические характеристики классического варианта сети *Token-Ring*:

- максимальное количество концентраторов типа IBM 8228 MAU – 12;
- максимальное количество абонентов в сети – 96;
- максимальная длина кабеля между абонентом и концентратором – 45 метров;
- максимальная длина кабеля между концентраторами – 45 метров;
- максимальная длина кабеля, соединяющего все концентраторы – 120 метров;
- скорость передачи данных – 4 Мбит/с и 16 Мбит/с.

Все приведенные характеристики относятся к случаю использования неэкранированной витой пары. Если применяется другая *среда передачи*,

характеристики сети могут отличаться. Например, при использовании экранированной витой пары (STP) количество абонентов может быть увеличено до 260 (вместо 96), длина кабеля – до 100 метров (вместо 45), количество концентраторов – до 33, а полная длина кольца, соединяющего концентраторы – до 200 метров. Оптоволоконный кабель позволяет увеличивать длину кабеля до двух километров.

Для передачи информации в *Token-Ring* применяется Манчестерское кодирование.

Для присоединения кабелей в *Token-Ring* используются разъемы RJ-45 (для неэкранированной витой пары), а также MIC и DB9P. Провода в кабеле соединяют одноименные контакты разъемов (то есть используются так называемые "прямые" кабели).

Сеть *Token-Ring* в классическом варианте уступает сети *Ethernet* как по допустимому размеру, так и по максимальному количеству абонентов. Что касается скорости передачи, то в настоящее время имеются версии *Token-Ring* на скорость 100 Мбит/с (High Speed Token-Ring, HSTR) и на 1000 Мбит/с (Gigabit Token-Ring). Компании, поддерживающие *Token-Ring* (среди которых IBM, Olicom, Madge), не намерены отказываться от своей сети, рассматривая ее как достойного конкурента *Ethernet*.

По сравнению с аппаратурой *Ethernet* аппаратура *Token-Ring* заметно дороже, так как используется более сложный *метод управления* обменом, поэтому сеть *Token-Ring* не получила столь широкого распространения.

Однако в отличие от *Ethernet* сеть *Token-Ring* значительно лучше держит высокий уровень нагрузки (более 30—40%) и обеспечивает гарантированное время доступа. Это необходимо, например, в сетях производственного назначения, в которых задержка реакции на внешнее событие может привести к серьезным авариям.

11. ТЕХНОЛОГИЯ FDDI

11.1. История создания стандарта FDDI

Технология Fiber Distributed Data Interface - первая технология локальных сетей, которая использовала в качестве среды передачи данных оптоволоконный кабель.

Недорогие оптические волокна, обеспечивающие низкие потери мощности светового сигнала и широкую полосу пропускания (до нескольких ГГц) появились только в 1970-е годы. В начале 1980-х годов началось промышленная установка и эксплуатация оптоволоконных каналов связи для территориальных телекоммуникационных систем.

В 1980-е годы начались также работы по созданию стандартных технологий и устройств для использования оптоволоконных каналов в локальных сетях. Работы по обобщению опыта и разработке первого оптоволоконного стандарта для локальных сетей были сосредоточены в Американском Национальном Институте по Стандартизации - ANSI, в рамках созданного для этой цели комитета X3T9.5.

В настоящее время большинство сетевых технологий поддерживают оптоволоконные кабели в качестве одного из вариантов физического уровня, но FDDI остается наиболее отработанной высокоскоростной технологией, стандарты на которую прошли проверку временем и устоялись, так что оборудование различных производителей показывает хорошую степень совместимости.

11.2. Основы технологии FDDI

Технология FDDI во многом основывается на технологии Token Ring, развивая и совершенствуя ее основные идеи. Разработчики технологии FDDI ставили перед собой в качестве наиболее приоритетных следующие цели:

- повысить битовую скорость передачи данных до 100 Мбит/с;
- повысить отказоустойчивость сети за счет стандартных процедур восстановления ее после отказов различного рода - повреждения кабеля,

некорректной работы узла, концентратора, возникновения высокого уровня помех на линии и т.п.;

- максимально эффективно использовать потенциальную пропускную способность сети как для асинхронного, так и для синхронного трафиков.

Сеть FDDI строится на основе двух оптоволоконных колец, которые образуют основной и резервный пути передачи данных между узлами сети. Использование двух колец - это основной способ повышения отказоустойчивости в сети FDDI, и узлы, которые хотят им воспользоваться, должны быть подключены к обоим кольцам. В нормальном режиме работы сети данные проходят через все узлы и все участки кабеля первичного (Primary) кольца, поэтому этот режим назван режимом *Thru* - "сквозным" или "транзитным". Вторичное кольцо (Secondary) в этом режиме не используется.

В случае какого-либо вида отказа, когда часть первичного кольца не может передавать данные (например, обрыв кабеля или отказ узла), первичное кольцо объединяется со вторичным (рисунок 11.1), образуя вновь единое кольцо. Этот режим работы сети называется *Wrap*, то есть "свертывание" или "сворачивание" колец. Операция свертывания производится силами концентраторов и/или сетевых адаптеров FDDI. Для упрощения этой процедуры данные по первичному кольцу всегда передаются против часовой стрелки, а по вторичному - по часовой.

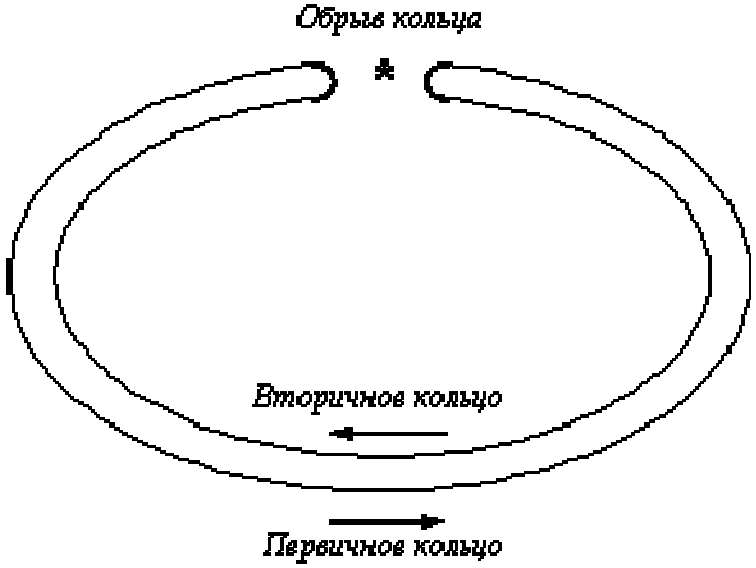


Рисунок 11.1. Реконфигурация колец FDDI при отказе

Кольца в сетях FDDI рассматриваются как общая разделяемая среда передачи данных, поэтому для нее определен специальный метод доступа. Этот метод очень близок к методу доступа сетей Token Ring и также называется методом маркерного кольца - token ring (рисунок 11.2, а).

Станция может начать передачу своих собственных кадров данных только в том случае, если она получила от предыдущей станции специальный кадр - маркер доступа (рисунок 11.2, б). После этого она может передавать свои кадры, если они у нее имеются, в течение времени, называемого временем удержания маркера - ***Token Holding Time (THT)***. После истечения времени THT станция обязана завершить передачу своего очередного кадра и передать маркер доступа следующей станции. Если же в момент принятия маркера у станции нет кадров для передачи по сети, то она немедленно транслирует маркер следующей станции.

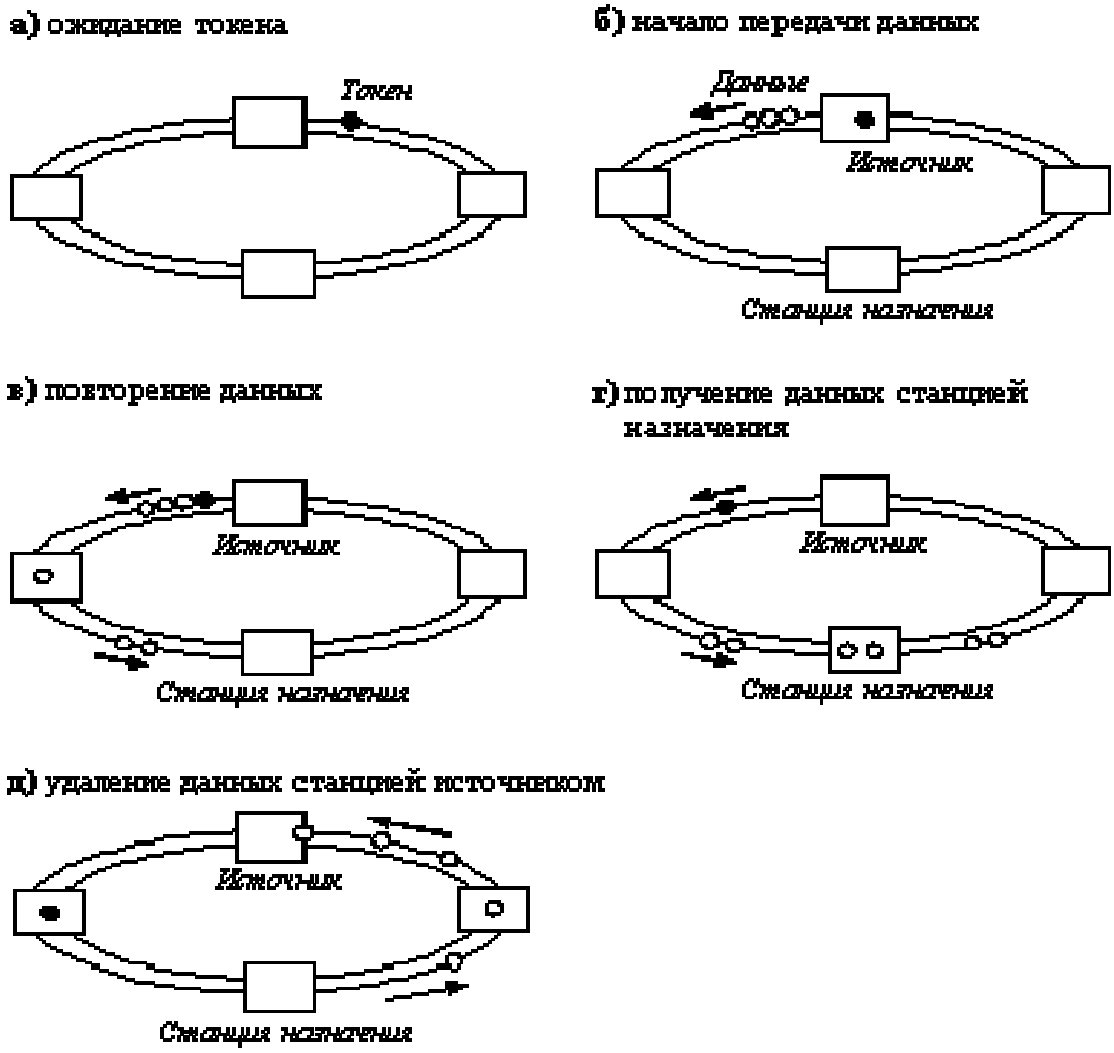


Рисунок 13.2. Обработка кадров станциями кольца FDDI

Каждая станция в сети постоянно принимает передаваемые ей предшествующим соседом кадры и анализирует их адрес назначения. Если адрес назначения не совпадает с ее собственным, то она транслирует кадр своему последующему соседу. Этот случай приведен на рисунке (рисунок 11.2, в). Нужно отметить, что, если станция захватила маркер и передает свои собственные кадры, то на протяжении этого периода времени она не транслирует приходящие кадры, а удаляет их из сети.

Если же адрес кадра совпадает с адресом станции, то она копирует кадр в свой внутренний буфер, проверяет его корректность (в основном по контрольной сумме), передает его поле данных для последующей обработки протоколу лежащего выше над FDDI уровня (например, IP), а затем передает исходный кадр по сети последующей станции (рисунок 13.1, г). В передаваемом в сеть кадре станция назначения отмечает три признака: распознавания адреса, копирования кадра и отсутствия или наличия в нем ошибок.

После этого кадр продолжает путешествовать по сети, транслируясь каждым узлом. Станция, являющаяся источником кадра для сети, ответственна за то, чтобы удалить кадр из сети, после того, как он, совершив полный оборот, вновь дойдет до нее (рисунок 11.2, д). При этом исходная станция проверяет признаки кадра, дошел ли он до станции назначения и не был ли при этом поврежден. Процесс восстановления информационных кадров не входит в обязанности протокола FDDI, этим должны заниматься протоколы более высоких уровней.

На рисунке 11.3 приведена структура протоколов технологии FDDI в сравнении с семиуровневой моделью OSI. FDDI определяет протокол физического уровня и протокол подуровня доступа к среде (MAC) канального уровня. Как и многие другие технологии локальных сетей, технология FDDI использует протокол 802.2 подуровня управления каналом данных (LLC), определенный в стандартах IEEE 802.2 и ISO 8802.2. FDDI использует первый тип процедур LLC, при котором узлы работают в дейтаграммном режиме - без

установления соединений и без восстановления потерянных или поврежденных кадров.

Физический уровень разделен на два подуровня: независимый от среды подуровень *PHY (Physical),* и зависящий от среды подуровень *PMD (Physical Media Dependent).* Работу всех уровней контролирует протокол управления станцией *SMT (Station Management).*

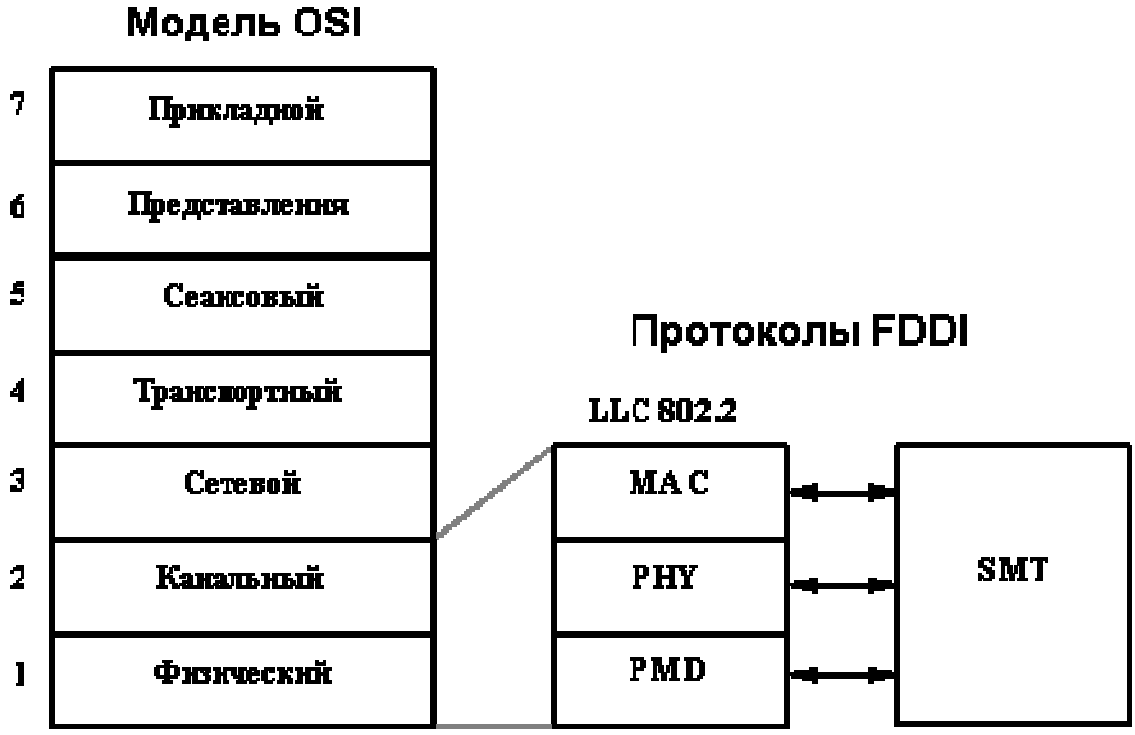


Рисунок 11.3. Структура протоколов технологии FDDI

Уровень PMD обеспечивает необходимые средства для передачи данных от одной станции к другой по оптоволокну. В его спецификации определяются:

- требования к мощности оптических сигналов и к многомодовому оптоволоконному кабелю 62.5/125 мкм;
- требования к оптическим обходным переключателям (optical bypass switches) и оптическим приемопередатчикам;
- параметры оптических разъемов MIC (Media Interface Connector), их маркировка;
- длина волны в 1300 нанометров, на которой работают приемопередатчики;
- представление сигналов в оптических волокнах в соответствии с методом NRZI.

Уровень PHY выполняет кодирование и декодирование данных, циркулирующих между MAC-уровнем и уровнем PMD, а также обеспечивает тактирование информационных сигналов. В его спецификации определяются:

- кодирование информации в соответствии со схемой 4B/5B;
- правила тактирования сигналов;
- требования к стабильности тактовой частоты 125 МГц;
- правила преобразования информации из параллельной формы в последовательную.

Уровень MAC ответственен за управление доступом к сети, а также за прием и обработку кадров данных. В нем определены следующие параметры:

- протокол передачи маркера;
- правила захвата и ретрансляции маркера;
- формирование кадра;
- правила генерации и распознавания адресов;
- правила вычисления и проверки 32-разрядной контрольной суммы.

Уровень SMT выполняет все функции по управлению и мониторингу всех остальных уровней стека протоколов FDDI. В управлении кольцом принимает участие каждый узел сети FDDI. Поэтому все узлы обмениваются специальными кадрами SMT для управления сетью. В спецификации SMT определено следующее:

- алгоритмы обнаружения ошибок и восстановления после сбоев;
- правила мониторинга работы кольца и станций;
- управление кольцом;
- процедуры инициализации кольца.

Отказоустойчивость сетей FDDI обеспечивается за счет управления уровнем SMT другими уровнями: с помощью уровня PHY устраняются отказы сети по физическим причинам, например, из-за обрыва кабеля, а с помощью уровня MAC - логические отказы сети, например, потеря нужного внутреннего пути передачи маркера и кадров данных между портами концентратора.

11.3. Типы узлов и правила их соединения в сеть

Все станции в сети FDDI делятся на несколько типов по следующим признакам:

- конечные станции или концентраторы;
- по варианту присоединения к первичному и вторичному кольцам;
- по количеству MAC-узлов и, соответственно, MAC-адресов у одной станции.

Одиночное и двойное присоединение к сети

Если станция присоединена только к первичному кольцу, то такой вариант называется одиночным присоединением - *Single Attachment, SA* (рисунок 11.4, а). Если же станция присоединена и к первичному, и ко вторичному кольцам, то такой вариант называется двойным присоединением - *Dual Attachment, DA* (рисунок 11.4, б).

Очевидно, что станция может использовать свойства отказоустойчивости, обеспечиваемые наличием двух колец FDDI, только при ее двойном подключении.

Как видно из рисунка 11.5, реакция станций на обрыв кабеля заключается в изменении внутренних путей передачи информации между отдельными компонентами станции.

Количество MAC-узлов у станции

Для того, чтобы иметь возможность передавать собственные данные в кольцо (а не просто ретранслировать данные соседних станций), станция должна иметь в своем составе хотя бы один MAC-узел, который имеет свой уникальный MAC-адрес. Станции могут не иметь ни одного узла MAC, и, значит, участвовать только в ретрансляции чужих кадров. Но обычно все станции сети FDDI, даже концентраторы, имеют хотя бы один MAC. Концентраторы используют MAC-узел для захвата и генерации служебных кадров, например, кадров инициализации кольца, кадров поиска неисправности в кольце и т.п.

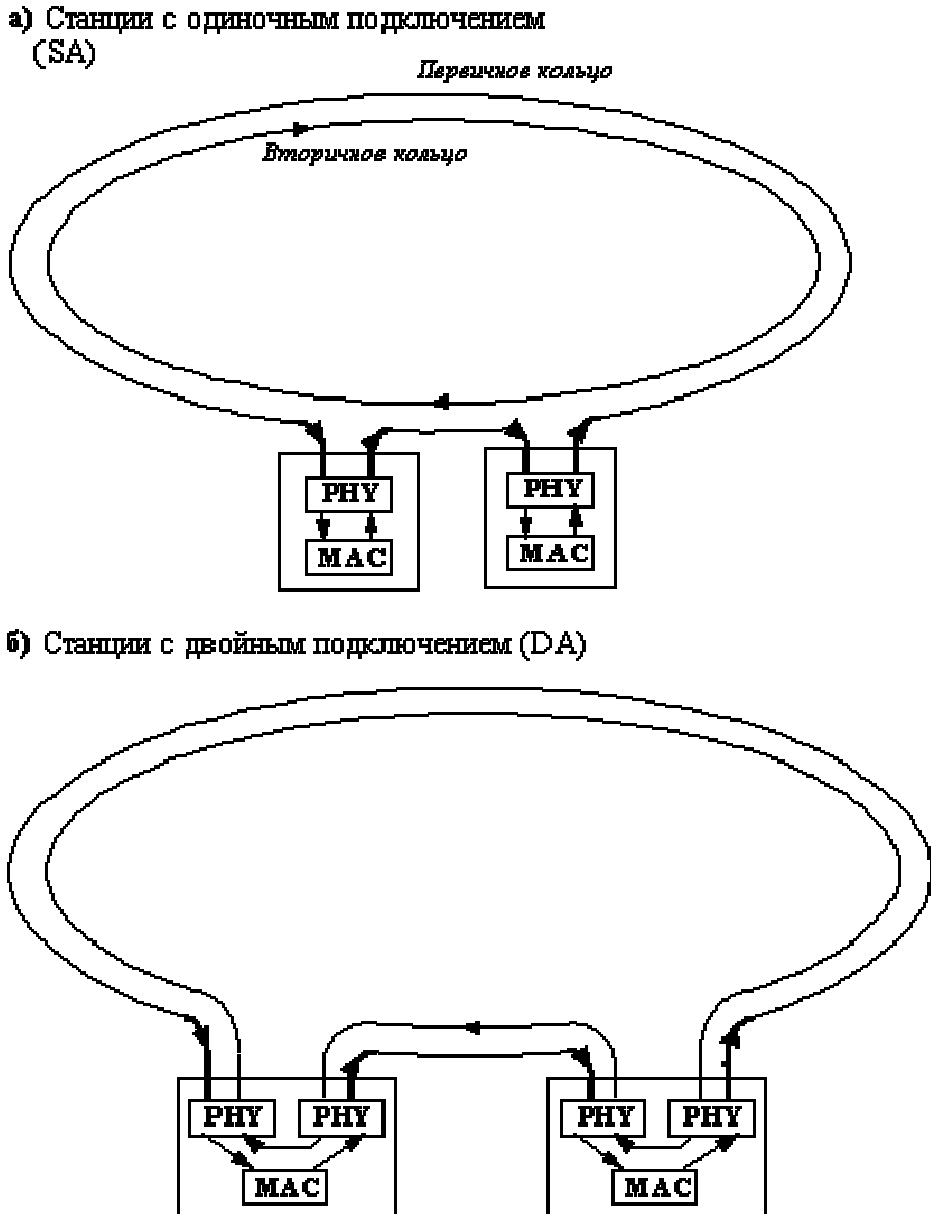


Рисунок 11.4. Одиночное (SA) и двойное (DA) подключение станций

Станции, которые имеют один MAC-узел, называются *SM (Single MAC)* станциями, а станции, которые имеют два MAC-узла, называются *DM (Dual MAC)* станциями.

В зависимости от того, является ли станция концентратором или конечной станцией, приняты следующие обозначения в зависимости от типа их подключения:

SAS (Single Attachment Station) - конечная станция с одиночным подключением,

DAS (Dual Attachment Station) - конечная станция с двойным подключением,

SAC (Single Attachment Concentrator) - концентратор с одиночным подключением,

DAC (Dual Attachment Concentrator) - концентратор с двойным подключением.

Типы портов станций и концентраторов FDDI и правила их соединения

В стандарте FDDI описаны четыре типа портов, которые отличаются своим назначением и возможностями соединения друг с другом для образования корректных конфигураций сетей.

На рисунке 11.5 показано типичное использование портов разных типов для подключения станций SAS и DAS к концентратору DAC.

Соединение портов S - S является допустимым, так как создает изолированное первичное кольцо, соединяющее только две станции, но обычно неиспользуемым.

Соединение портов M - M является запрещенным, а соединения A-A, B-B, A-S, S-A, B-S, S-B - нежелательными, так как создают неэффективные комбинации колец.

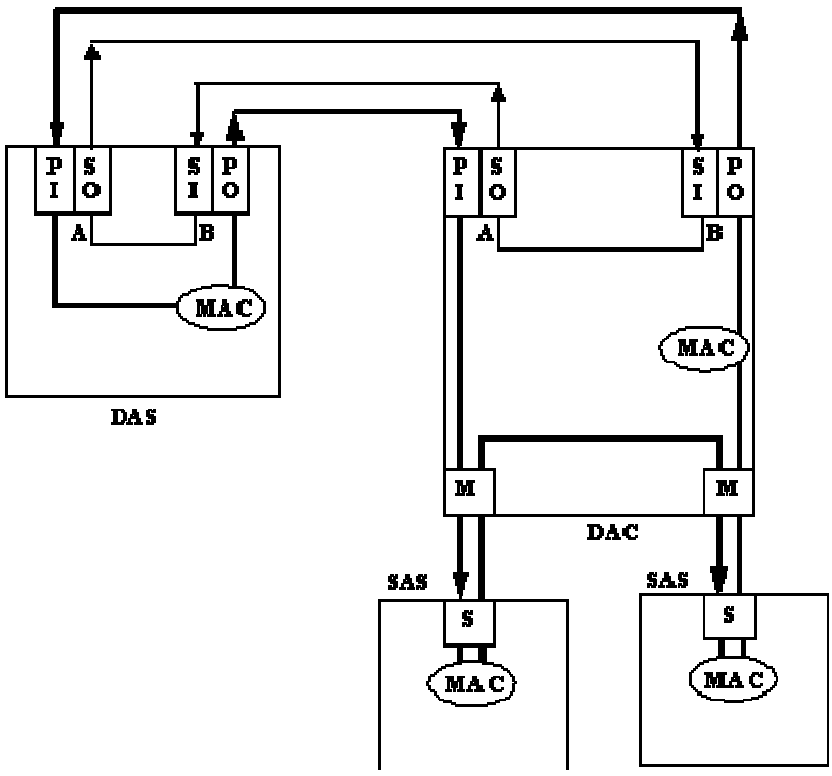


Рисунок 11.6. Использование портов различных типов

11.4. Спецификация зависящего от среды физического подуровня PMD

Структура физического соединения

Рассмотрим физический подуровень *PMD (Physical Media Dependent layer)*, определенный в стандарте FDDI для оптоволокна - Fiber PMD.

Эта спецификация определяет аппаратные компоненты для создания физических соединений между станциями: оптические передатчики, оптические приемники, параметры кабеля, оптические разъемы. Для каждого из этих элементов указываются конструктивные и оптические параметры, позволяющие станциям устойчиво взаимодействовать на определенных расстояниях.

Физическое соединение - основной строительный блок сети FDDI. Типичная структура физического соединения представлена на рисунке 11.6.

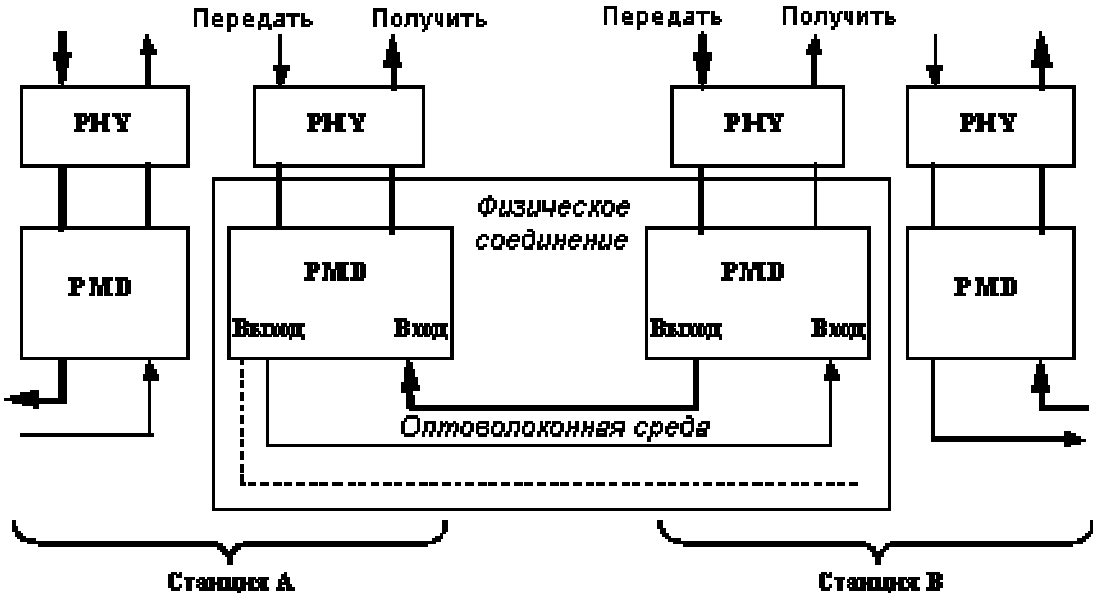


Рисунок 11.6. Физическое соединение сети FDDI

Каждое физическое соединение состоит из двух физических связей - первичной и вторичной. Эти связи являются односторонними - данные передаются от передатчика одного устройства PHY к приемнику другого устройства PHY.

Кабели и разъемы

Основной вид кабеля для стандарта Fiber PMD - многомодовый кабель с диаметром сердечника 62.5 мкм и диаметром отражающей оболочки 125 мкм.

Кроме основного вида кабеля, спецификация Fiber PMD допускает использование многомодовых кабелей с диаметром сердечника в 50 мкм, 85 мкм и 100 мкм.

В качестве разъемов стандарт Fiber PMD определяет оптические разъемы *MIC (Media Interface Connector).* Разъем MIC обеспечивает подключение 2-х волокон кабеля, соединенных с вилкой MIC, к 2-м волокнам порта станции, соединенными с розеткой MIC. Стандартизованы только конструктивные параметры розетки MIC, а любые вилки MIC, подходящие к стандартным розеткам MIC, считаются пригодными к использованию.

Кроме разъемов MIC, допускается использовать разъемы ST и SC, выпускаемые промышленностью.

В качестве источника света допускается использование светодиодов (LED) или лазерных диодов с длиной волны 1.3 мкм.

Кроме многомодового кабеля, допускается использование более качественного одномодового кабеля *(Single Mode Fiber, SMF)* и разъемов SMF-MIC для этого кабеля. В этом случае дальность физического соединения между соседними узлами может увеличиться до 40 км - 60 км, в зависимости от качества кабеля, разъемов и соединений. Требования, определенные в спецификации SMF-PMD, для мощности на выходе передатчика и входе приемника, те же, что и для одномодового кабеля.

11.5. Физический подуровень PHY

Если в задачи подуровня PMD входит формирование качественных оптических импульсов на выходе и входе каждого физического соединения, то подуровень PHY имеет дело с передачей с помощью импульсов PMD логических единиц и нулей, приходящих с подуровня MAC. Более точно, подуровень PHY занимается следующими задачами:

- определение моментов времени снятия информации по сигналам, поступающим от подуровня PMD (тактирование входных сигналов);
- определение границ байт при обмене данными с MAC-подуровнем;
- кодирование поступающих от MAC-подуровня символов в соответствующий физический код (NRZI или MLT-3) подуровня PMD;
- декодирование поступающих от PMD сигналов (NRZI или MLT-3) в символы MAC-подуровня;
- управление эластичным буфером (Elasticity Buffer) для согласования частоты входных и выходных сигналов;
- определение статуса входящей физической линии на основе тестовой последовательности управляющих символов;
- генерация последовательности управляющих символов для выходящей физической линии по командам от подуровня SMT;
- фильтрация приходящих ошибочных символов для исключения их передачи на выходную линии.

11.6. MAC-уровень

Функции MAC-уровня

В соответствии со стандартами IEEE 802 канальный уровень в локальных сетях состоит из двух подуровней - LLC и MAC. Стандарт FDDI не вводит свое определение подуровня LLC, а использует его сервисы, описанные в документе IEEE 802.2 LLC.

Подуровень MAC выполняет в технологии FDDI следующие функции:

- поддерживает сервисы для подуровня LLC;
- формирует кадр определенного формата;
- управляет процедурой передачи токена;
- управляет доступом станции к среде;
- адресует станции в сети;

- копирует кадры, предназначенные для данной станции, в буфер и уведомляет подуровень LLC и блок управления станцией SMT о прибытии кадра;
- генерирует контрольную последовательность кадра (CRC) и проверяет ее у всех кадров, циркулирующих по кольцу;
- удаляет из кольца все кадры, которые сгенерировала данная станция;
- управляет таймерами, которые контролируют логическую работу кольца - таймером удержания токена, таймером оборота токена и т.д.;
- ведет ряд счетчиков событий, что помогает обнаружить и локализовать неисправности;
- определяет механизмы, используемые кольцом для реакции на ошибочные ситуации - повреждение кадра, потерю кадра, потерю токена и т.д.

Операции MAC-уровня

Захват маркера. Если станция имеет право захватить маркер, то она после ретрансляции на выходной порт символов PA и SD маркера, удаляет из кольца символ FC, по которому она распознала маркер, а также конечный ограничитель ED. Затем она передает вслед за уже переданным символом SD символы своего кадра, таким образом, формируя его из начальных символов маркера.

Передача кадра. После удаления полей FC и ED токена станция начинает передавать символы кадров, которые ей предоставил для передачи уровень LLC. Станция может передавать кадры до тех пор, пока не истечет время удержания токена.

Для сетей FDDI предусмотрена передача кадров двух типов трафика - синхронного и асинхронного.

Синхронный трафик предназначен для приложений, которые требуют предоставления им гарантированной пропускной способности для передачи голоса, видеоизображений, управления процессами и других случаев работы в реальном времени. Для такого трафика каждой станции предоставляется фиксированная часть пропускной способности кольца FDDI, поэтому станция

имеет право передавать кадры синхронного трафика всегда, когда она получает маркер от предыдущей станции.

Асинхронный трафик - это обычный трафик локальных сетей, не предъявляющий высоких требований к задержкам обслуживания. Станция может передавать асинхронные кадры только в том случае, если при последнем обороте маркера по кольцу для этого осталась какая-либо часть неизрасходованной пропускной способности. Интервал времени, в течение которого станция может передавать асинхронные кадры, называется временем удержания маркера *(Token Holding Time, THT).* Каждая станция самостоятельно вычисляет текущее значение этого параметра по алгоритму, рассмотренному ниже.

Станция прекращает передачу кадров в двух случаях: либо при истечении времени удержания маркера THT, либо при передаче всех имеющихся у нее кадров до истечения этого срока. После передачи последнего своего кадра станция формирует маркер и передает его следующей станции.

Обработка кадра станцией назначения. Станция назначения, распознав свой адрес в поле DA, начинает копировать символы кадра во внутренний буфер одновременно с повторением их на выходном порту. При этом станция назначения устанавливает признак распознавания адреса. Если же кадр скопирован во внутренний буфер, то устанавливается и признак копирования (невыполнение копирования может произойти, например, из-за переполнения внутреннего буфера). Устанавливается также и признак ошибки, если ее обнаружила проверка по контрольной последовательности.

Удаление кадра из кольца. Каждый MAC-узел ответственен за удаление из кольца кадров, которые он ранее в него поместил. Этот процесс известен под названием *Frame Stripping*. Если MAC-узел при получении своего кадра занят передачей следующих кадров, то он удаляет все символы вернувшегося по кольцу кадра. Если же он уже освободил маркер, то он повторяет на выходе несколько полей этого кадра прежде, чем распознает свой адрес в поле SA. В этом случае в кольце возникает усеченный кадр, у которого после поля SA следуют символы

Idle и отсутствует конечный ограничитель. Этот усеченный кадр будет удален из кольца какой-нибудь станцией, принявшей его в состоянии собственной передачи.

11.7.Общая характеристика функций управления сетью по спецификации SMT

Кроме спецификаций уровней PHY, PMD и MAC, стандарт FDDI определяет также спецификацию уровня управления станцией Station Management (SMT).

Эта спецификация определяет функции, которые должен выполнять каждый узел в сети FDDI. SMT контролирует и управляет всеми процессами канального и физического уровней, протекающими в отдельной станции. Кроме того, процесс SMT каждой станции взаимодействует с аналогичными процессами других станций для того, чтобы следить и координировать все операции в кольце FDDI. В этом случае SMT принимает участие в распределенном одноранговом управлении кольцом.

SMT включает три группы функций:

- Управление соединениями - Connection Management (CMT);
- Управление кольцом - Ring Management (RMT);
- Управление, основанное на кадрах - Frame-Based Management (FBM).

Функции *управления соединениями CMT* уже были рассмотрены ранее в связи с тем, что их основным назначением является контроль и управление физическими соединениями, организуемыми физическим уровнем.

Функции *управления кольцом RMT* заключаются в управлении локальными узлами MAC и кольцами, к которым они присоединены. Функции RMT ответственны за обнаружение дублированных адресов, а также за запуск процедуры инициации кольца Claim Token и процедур обработки аварийных ситуаций Beacon и Trace.

Функции *управления, основанного на кадрах **FBM*** позволяют узлу получать от других узлов сети информацию о их состоянии и статистике о прошедшем через них трафике. Эта информация хранится в базе данных управляющей информации MIB (Management Information Base).

12.ТЕХНОЛОГИЯ 100VG-AnyLAN

12.1. Общая характеристика технологии 100VG-AnyLAN

В качестве альтернативы технологии Fast Ethernet, фирмы AT&T и HP выдвинули проект новой технологии со скоростью передачи данных 100 Мбит/с - ***100Base-VG***. В этом проекте было предложено усовершенствовать метод доступа с учетом потребности мультимедийных приложений, при этом сохранить совместимость формата пакета с форматом пакета сетей 802.3. В сентябре 1993 года по инициативе фирм IBM и HP был образован комитет IEEE 802.12, который занялся стандартизацией новой технологии. Проект был расширен за счет поддержки в одной сети кадров не только формата Ethernet, но и формата Token Ring. В результате новая технология получила название ***100VG-AnyLAN,*** то есть технология для любых сетей (Any LAN - любые сети), имея в виду, что в локальных сетях технологии Ethernet и Token Ring используются в подавляющем количестве узлов.

Летом 1995 года технология 100VG-AnyLAN получила статус стандарта IEEE 802.12.

В технологии 100VG-AnyLAN определены новый метод доступа Demand Priority и новая схема квартетного кодирования Quartet Coding, использующая избыточный код 5B/6B.

Метод доступа ***Demand Priority*** основан на передаче концентратору функций арбитра, решающего проблему доступа к разделяемой среде. Метод Demand Priority повышает коэффициент использования пропускной способности сети за счет введения простого, детерминированного метода разделения общей

среды, использующего два уровня приоритетов: низкий - для обычных приложений и высокий - для мультимедийных.

14.2. Структура сети 100VG-AnyLAN

Сеть 100VG-AnyLAN всегда включает центральный концентратор, называемый концентратором уровня 1 или корневым концентратором (рис.12.1).

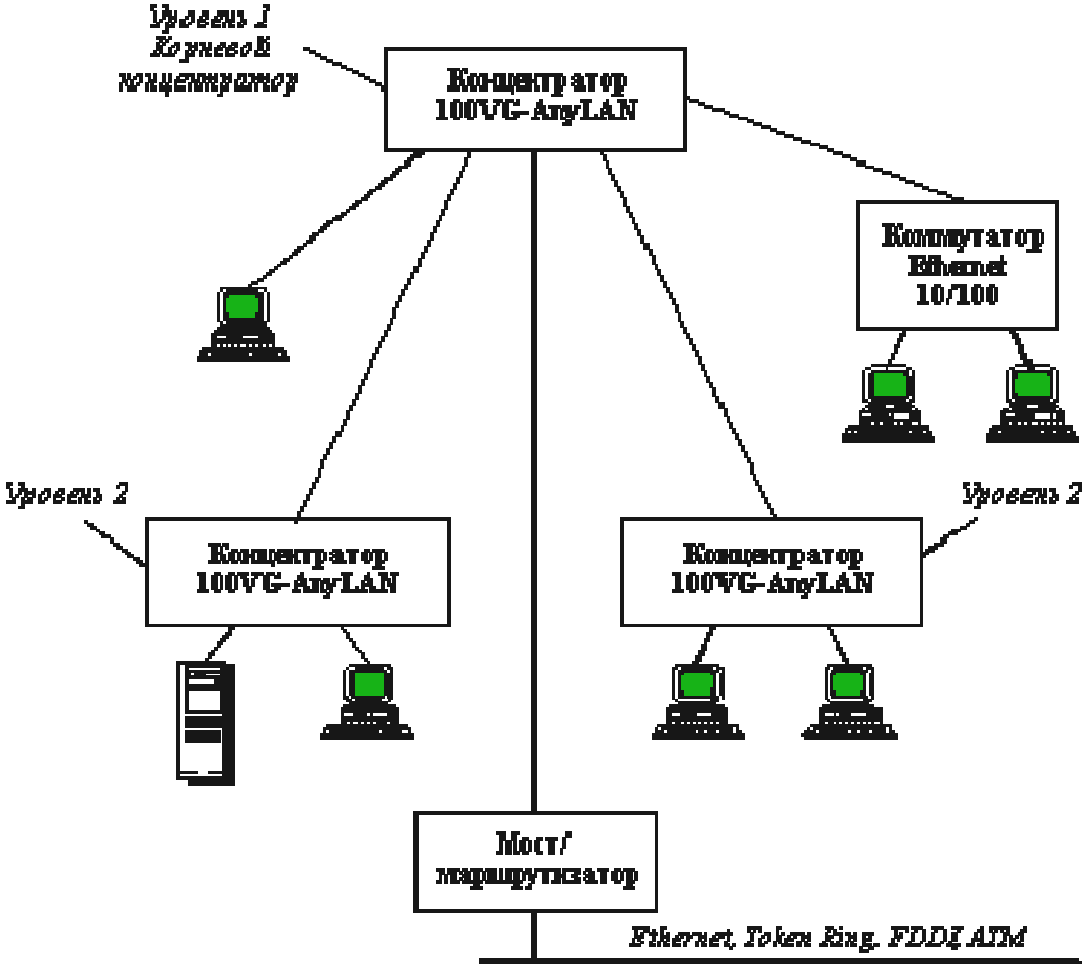


Рисунок 12.1 . Структура сети 100VG-AnyLAN

Корневой концентратор имеет связи с каждым узлом сети, образуя топологию типа звезда. Этот концентратор представляет собой интеллектуальный центральный контроллер, который управляет доступом к сети, постоянно выполняя цикл "кругового" сканирования своих портов и проверяя наличие запросов на передачу кадров от присоединенных к ним узлов. Концентратор принимает кадр от узла, выдавшего запрос, и передает его только через тот порт,

к которому присоединен узел с адресом, совпадающим с адресом назначения, указанным в кадре.

Каждый концентратор может быть сконфигурирован на поддержку либо кадров 802.3 Ethernet, либо кадров 802.5 Token Ring. Все концентраторы, расположенные в одном и том же логическом сегменте (не разделенном мостами, коммутаторами или маршрутизаторами), должны быть сконфигурированы на поддержку кадров одного типа. Для соединения сетей 100VG-AnyLAN, использующих разные форматы кадров 802.3, нужен мост, коммутатор или маршрутизатор.

Каждый концентратор имеет один "восходящий" (up-link) порт и N "нисходящих" портов (down-link), как это показано на рисунке 12.2.

Восходящий порт работает как порт узла, но он зарезервирован для присоединения в качестве узла к концентратору более высокого уровня. Нисходящие порты служат для присоединения узлов, в том числе и концентраторов нижнего уровня. Каждый порт концентратора может быть сконфигурирован для работы в нормальном режиме или в режиме монитора. Порт, сконфигурированный для работы в нормальном режиме, передает только те кадры, которые предназначены узлу, подключенному к данному порту. Порт, сконфигурированный для работы в режиме монитора, передает все кадры, обрабатываемые концентратором.

Узел представляет собой компьютер или коммуникационное устройство технологии 100VG-AnyLAN - мост, коммутатор, маршрутизатор или концентратор. Концентраторы, подключаемые как узлы, называются концентраторами 2-го и 3-го уровней. Всего разрешается образовывать до трех уровней иерархии концентраторов.

Связь, соединяющая концентратор и узел, может быть образована либо 4 парами неэкранированной витой пары категорий 3, 4 или 5 (4-UTP Cat 3, 4, 5), либо 2 парами неэкранированной витой пары категории 5 (2-UTP Cat 5), либо 2 парами экранированной витой пары типа 1 (2-STP Type 1), либо 2 парами многомодового оптоволоконного кабеля.

Варианты кабельной системы могут использоваться любые, но ниже будет рассмотрен вариант 4-UTP, который был разработан первым и получил наибольшее распространение.

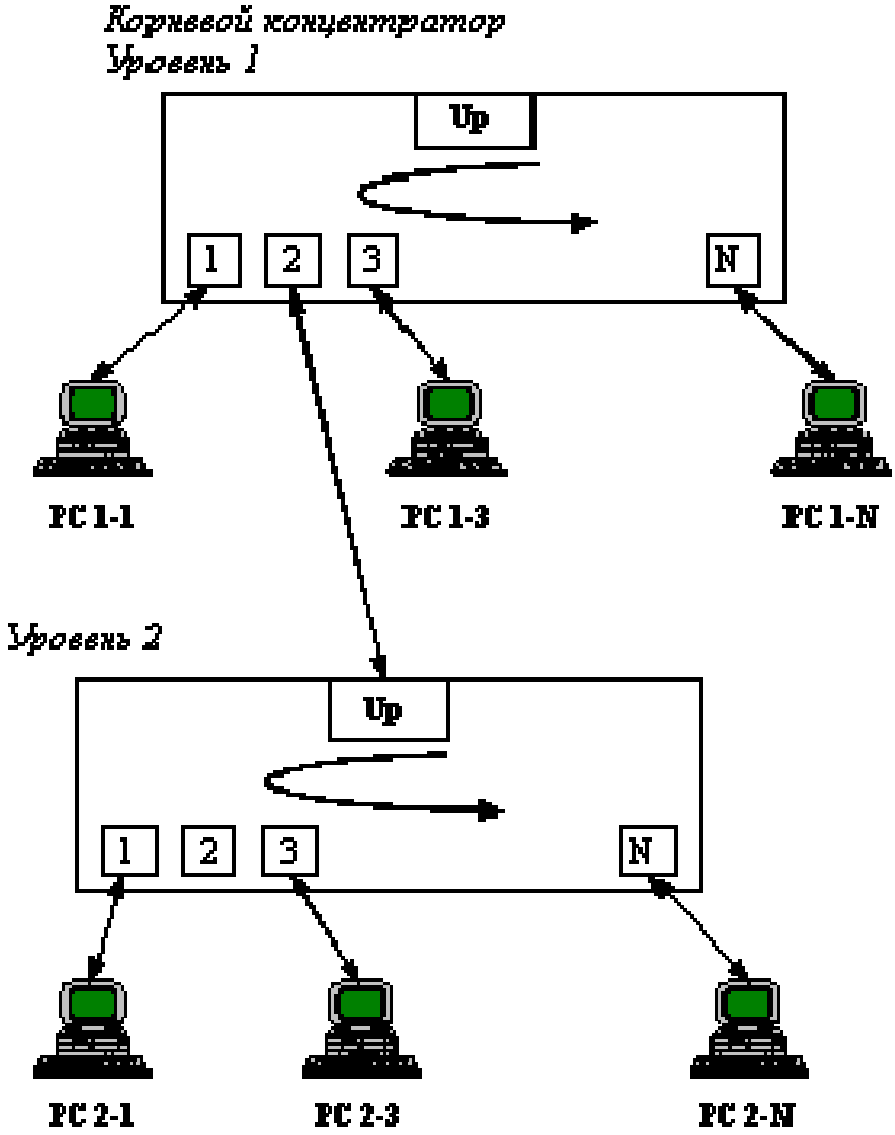


Рисунок 12.2. Круговой опрос портов концентраторами сети 100VG-AnyLAN

12.3. Стек протоколов технологии 100VG-AnyLAN

Структура стека протоколов технологии 100VG-AnyLAN согласуется с архитектурными моделями OSI/ISO и IEEE, в которых канальный уровень разделен на подуровни. Как видно из рисунка 12.3, стек протоколов технологии 100VG-AnyLAN состоит из подуровня доступа к среде *(Media Access Control, MAC),* подуровня, независящего от физической среды *(Physical Media Independent,*

PMI) и подуровня, зависящего от физической среды *(Physical Media Dependent, PMD).*

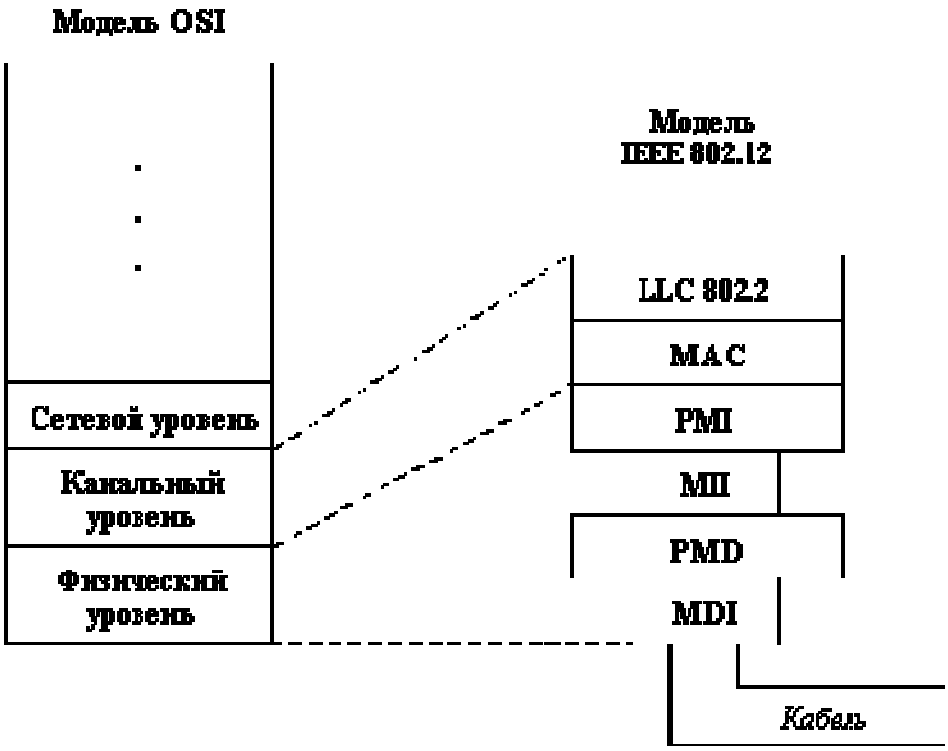


Рисунок 12.3. Структура стека протоколов технологии 100VG-AnyLAN

12.4. Функции уровня MAC

Функции уровня MAC включают реализацию протокола доступа Demand Priority, подготовки линии связи и формирования кадра соответствующего формата.

Метод Demand Priority (приоритетный доступ по требованию) основан на том, что узел, которому нужно передать кадр по сети, передает запрос (требование) на выполнение этой операции концентратору. Каждый запрос может иметь либо низкий, либо высокий приоритеты. Высокий приоритет отводиться для трафика чувствительных к задержкам мультимедийных приложений.

Высокоприоритетные запросы всегда обслуживаются раньше низкоприоритетных. Требуемый уровень приоритета кадра устанавливается протоколами верхних уровней, не входящими в технологию 100VG-AnyLAN и передается для отработки уровню MAC.

Как показано на рисунке 12.2, концентратор уровня 1 постоянно сканирует запросы узлов, используя алгоритм кругового опроса (round-robin). Это сканирование позволяет концентратору определить, какие узлы требуют передачи кадров через сеть и каковы их приоритеты.

В течение одного цикла кругового сканирования каждому узлу разрешается передать один кадр данных через сеть. Концентраторы, присоединенные как узлы к концентраторам верхних уровней иерархии, также выполняют свои циклы сканирования и передают запрос на передачу кадров концентратору. Концентратор нижнего уровня с N портами имеет право передать N кадров в течение одного цикла опроса.

Каждый концентратор ведет отдельные очереди для низкоприоритетных и высокоприоритетных запросов. Низкоприоритетные запросы обслуживаются только до тех пор, пока не получен высокоприоритетный запрос. В этом случае текущая передача низкоприоритетного кадра завершается и обрабатывается высокоприоритетный запрос. Перед возвратом к обслуживанию низкоприоритетных кадров должны быть обслужены все высокоприоритетные запросы. Для того чтобы гарантировать доступ для низкоприоритетных запросов в периоды высокой интенсивности поступления высокоприоритетных запросов, вводится порог ожидания запроса. Если у какого-либо низкоприоритетного запроса время ожидания превышает этот порог, то ему присваивается высокий приоритет.

На рисунке 12.2 показан пример цикла кругового опроса. Сначала предположим, что все порты передали запросы нормального приоритета, и что в начальный момент времени корневой концентратор начал круговой опрос. Порядок обслуживания портов будет следующим: 1-1 (уровень 1 - порт 1), 2 -1, 2-3, 2-N, 1-3, 1-N.

Теперь предположим, что узлы 1-1, 2-3 и 1-3 выставили высокоприоритетные запросы. В этом случае порядок обслуживания будет таким: 1-1, 2-3, 1-3, 2-1, 2-N, 1-N.

Уровень MAC получает кадр от уровня LLC и добавляет к нему адрес узла-источника, дополняет поле данных байтами-заполнителями до минимально допустимого размера, если это требуется, а затем вычисляет контрольную сумму и помещает ее в соответствующее поле. После этого кадр передается на физический уровень.

14.5. Функции уровня PMI

Функции, не зависящие от физической среды, представленные на рисунке 12.4, включают квартетную канальную шифрацию, кодирование 5B/6B, добавление к кадру преамбулы, начального и конечного ограничителей и передачу кадра на уровень PMD.

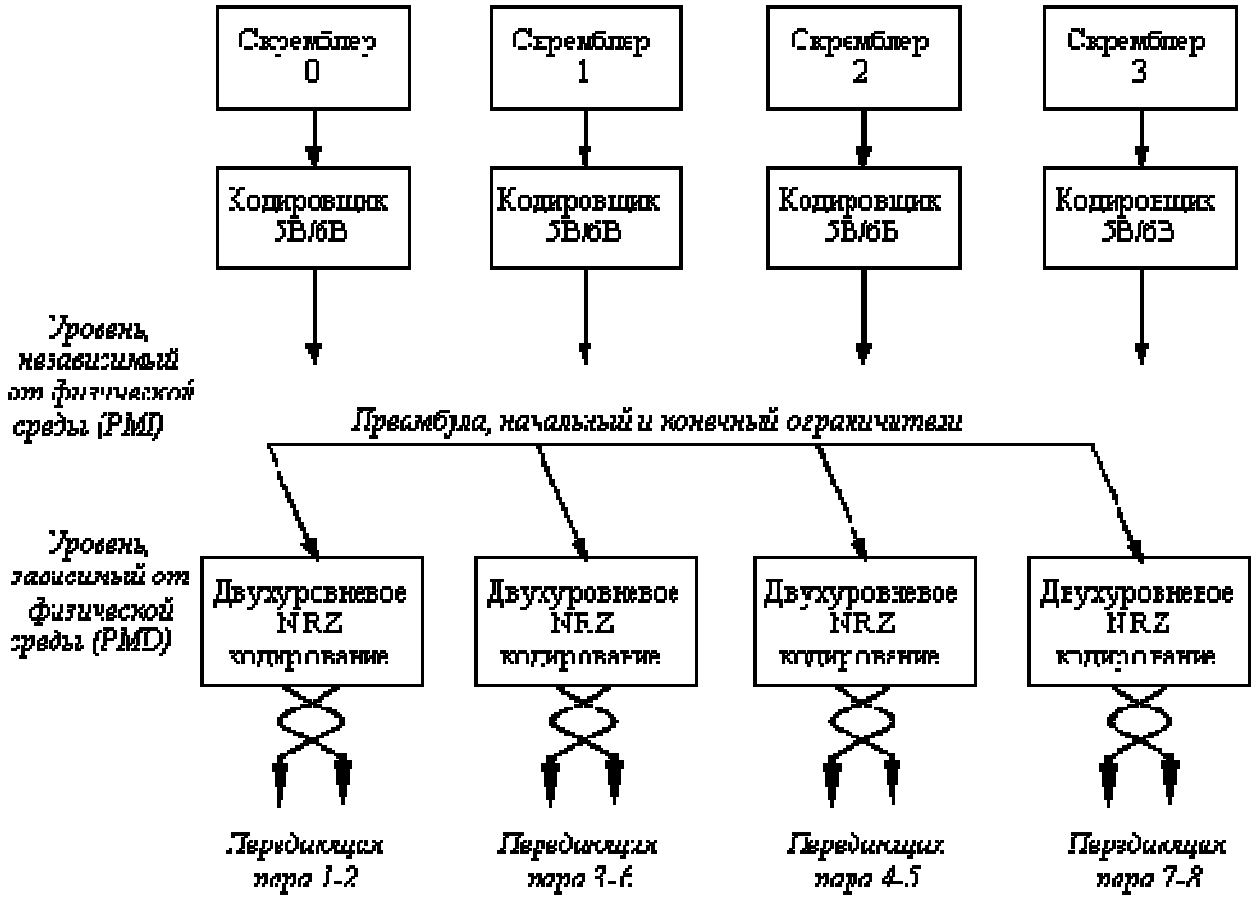


Рисунок 12.4. Функции уровней PMI и PMD

Процесс квартетного распределения по каналам состоит в последовательном делении байтов MAC-кадра на порции данных по 5 бит

(квинтеты), а также в последовательном распределении этих порций между четырьмя каналами, как это показано на рисунке 12.5.

Каждый из 4-х каналов представляет собой одну витую пару: канал 0 - пару, образованную контактами 1 и 2, канал 1 - пару 3 - 6, канал 2 - пару 4 - 5, канал 3 - пару 7 - 8. Двухпарные спецификации физического уровня PMD используют затем схему мультиплексирования, преобразующую 4 канала в 2 или 1.

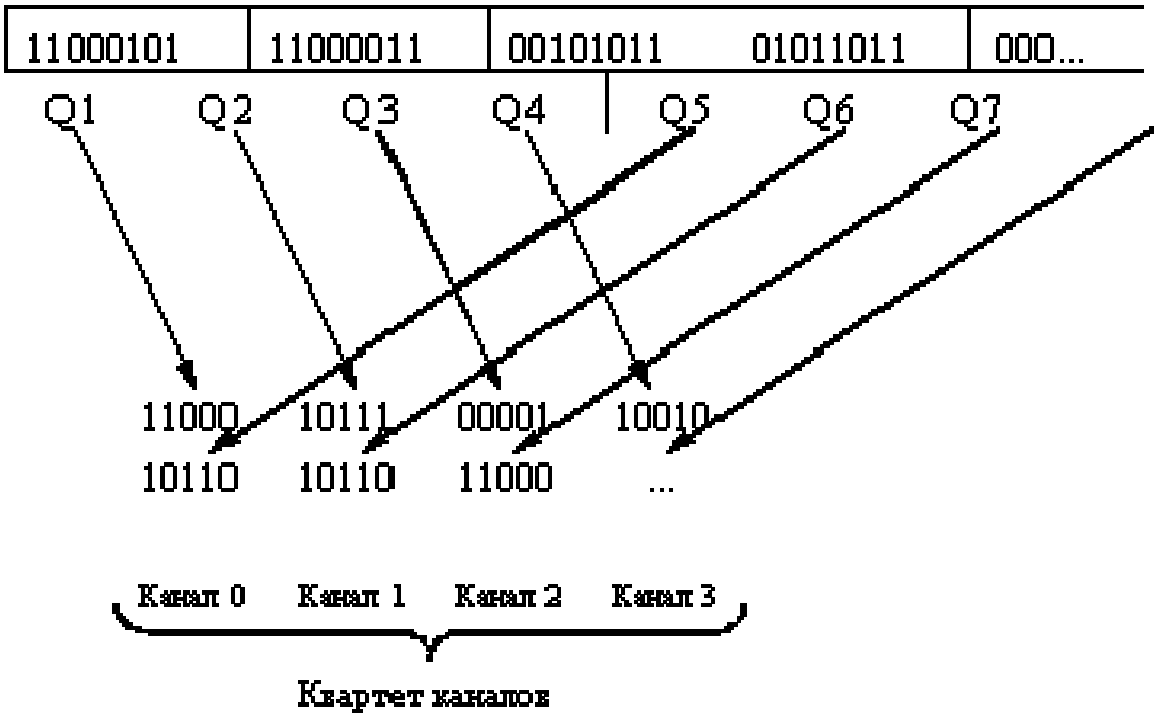


Рисунок 12.5. Распределение квинтетов по 4-м каналам

Шифрация данных состоит в случайном "перемешивании" квинтетов данных с целью исключения комбинаций из повторяющихся единиц или нулей. Перемешивание производится с помощью специальных устройств - скремблеров. Случайные наборы цифр уменьшают излучение радиоволн и взаимные наводки в кабеле.

Кодирование по схеме 5B/6B - это процесс отображения "перемешанных" квинтетов в заранее определенные 6-битовые коды. Этот процесс создает сбалансированные коды, содержащие равное количество единиц и нулей, что обеспечивает гарантированную синхронизацию приемника при изменениях входного сигнала.

Кодирование 5B/6B обеспечивает также контроль за ошибками при передаче, так как некорректные квинтеты, содержащие больше трех единиц или больше трех нулей, легко обнаружить.

На рисунке 12.6 приведен пример квинтетов данных, зашифрованных и преобразованных в символы 5B/6B. Поскольку существует только 16 сбалансированных символов, 32 комбинации, содержащиеся в квинтете, используют для своего представления два 6-ти битных символа, используемых по очереди для соблюдения баланса постоянного тока.

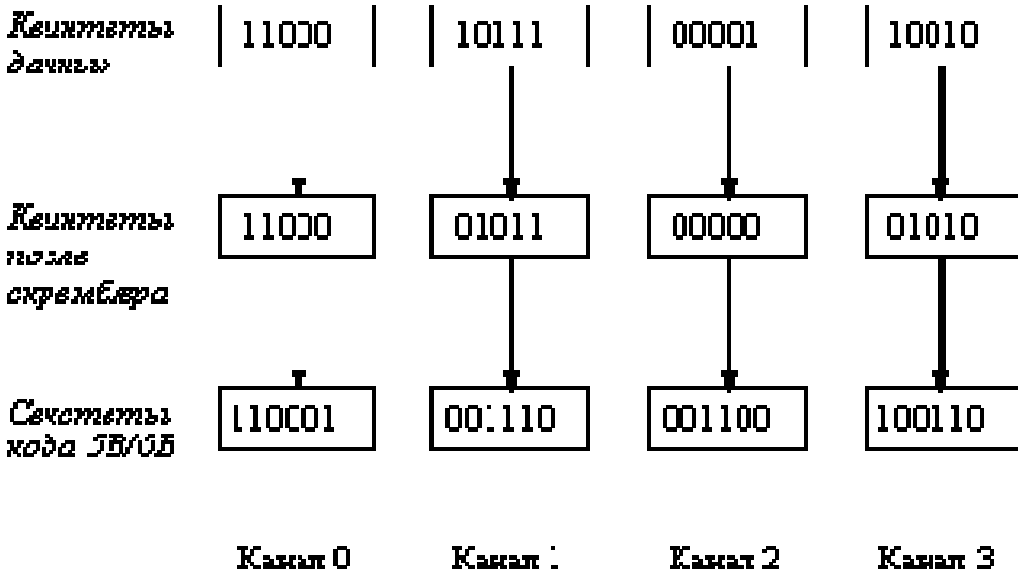


Рисунок 12.6. Пример шифрации и кодирования квинтетов

Преамбула, начальный и конечный ограничители добавляются в каждом канале для корректной передачи данных через сеть.

14.6. Функции уровня PMD

Функции зависимого от физической среды уровня PMD включают: мультиплексирование каналов (только для 2-х витых пар или оптоволокна), кодирование NRZ, операции передачи сигналов по среде и контроль статуса физической связи.

Технология 100VG-AnyLAN поддерживает следующие типы физической среды:

- 4-парную неэкранированную витую пару;

- 2-парную неэкранированную витую пару;
- 2-парную экранированную витую пару;
- одномодовый или многомодовый оптоволоконный кабель.

Далее будут рассмотрены детали спецификации PMD для 4-парной неэкранированной витой пары.

Рисунок 12.7 иллюстрирует применения NRZ кодирования, использующего для представления единиц потенциал высокого уровня, а для представления нулей - потенциал низкого уровня.

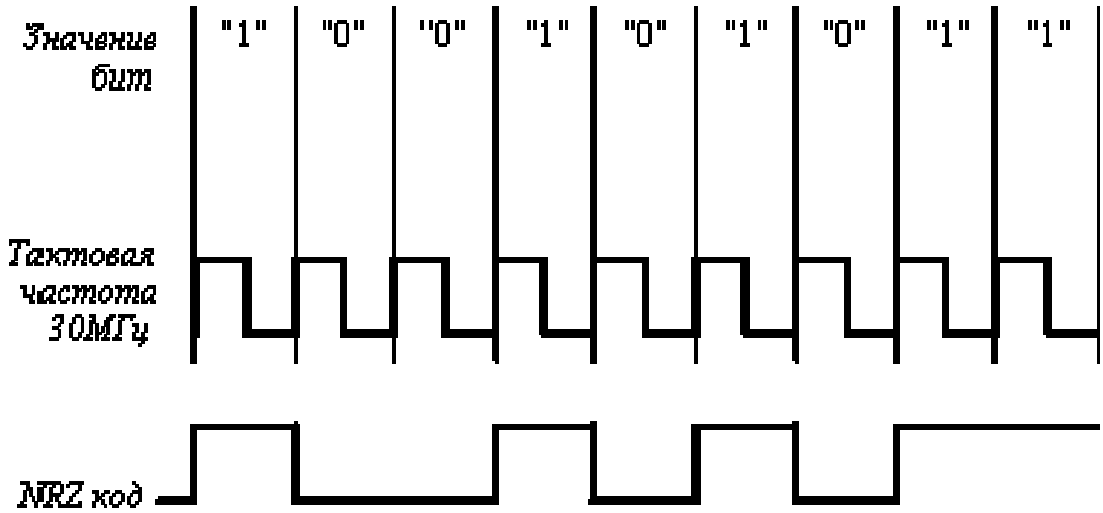


Рисунок 12.7. NRZ кодирование

Спецификация 4UTP, использующая 4-парную неэкранированную витую пару, использует тактовый генератор с частотой 30 МГц для передачи данных со скоростью 30 Мбит/с по каждому из четырех каналов, что в сумме дает 120 Мбит/с кодированных данных. Приемник получает кодированные данные со скоростью 30 Мбит/с по каждому каналу и преобразует их в поток исходных данных со скоростью 25 Мбит/с, что в результате дает пропускную способность в 100 Мбит/с.

Использованный метод представления данных в кабеле позволяет технологии 100VG-AnyLAN работать на голосовом кабеле (Voice-Grade) категории 3. Максимальная частота результирующего сигнала на кабеле не превышает 15 МГц, так как метод NRZ очень эффективен в отношении спектра сигналов. При тактовой частоте в 30 МГц частота 15 МГц генерируется только

при передаче кодов 10101010, что является для спектра результирующего сигнала наихудшим случаем. При передаче других кодов частота сигнала будет ниже 15 МГц.

Операции передачи данных на 4-парном кабеле используют как полнодуплексный, так и полудуплексный режимы (рисунок 12.8).

Полнодуплексные операции используются для одновременной передачи в двух направлениях - от узла к концентратору и от концентратора к узлу - сигнальной информации о состоянии линии. Сигнальная информация от концентратора идет по парам 1-2 и 3-6, а сигнальная информация от узла идет по парам 4-5 и 7-8.

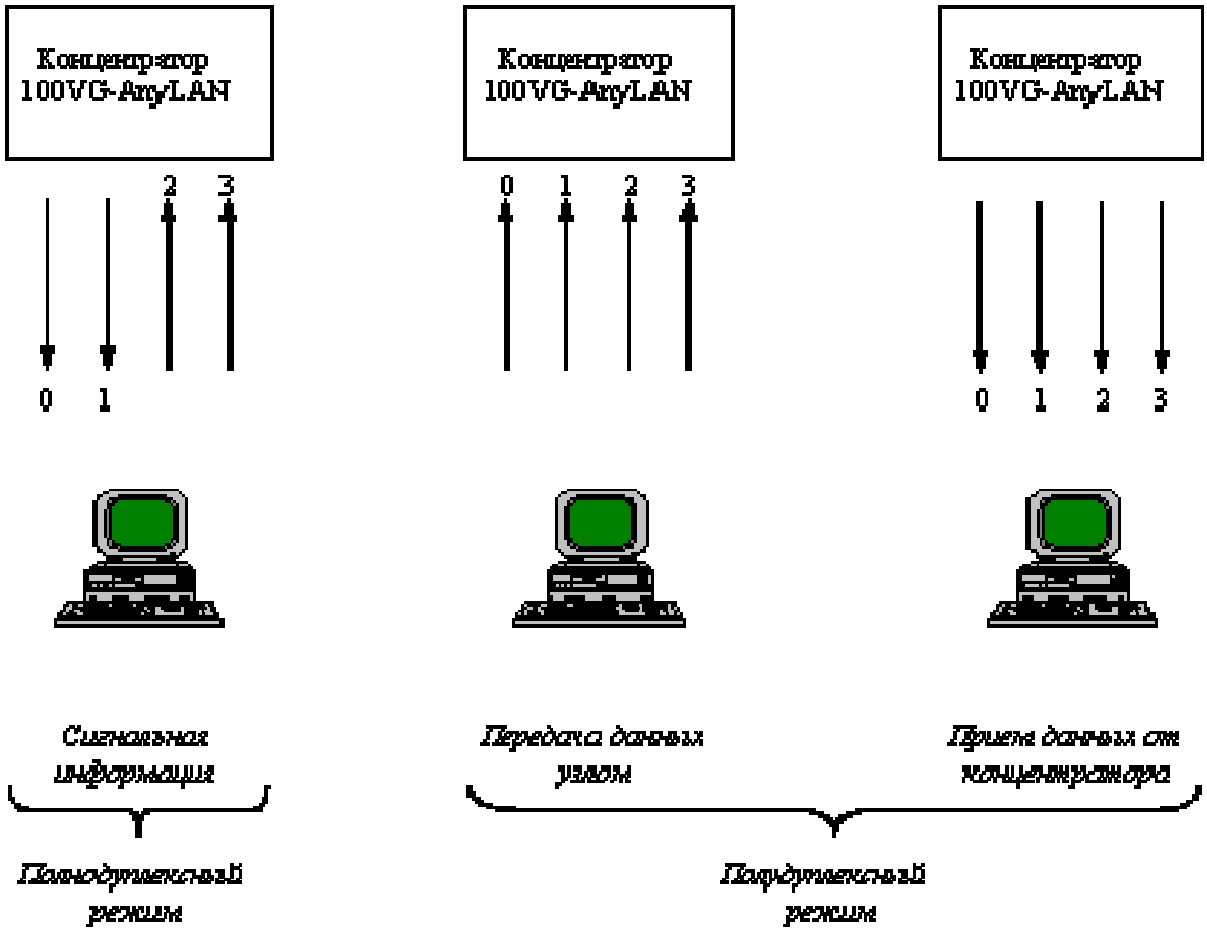


Рисунок 12.8. Полнодуплексные и полудуплексные операции

Полудуплексные операции используются для передачи данных от концентратора узлу и от узла концентратору по всем четырем парам.

Сигнализация о статусе связи, осуществляемая в полнодуплексном режиме, использует два низкочастотных сигнала, обозначаемые как тон 1 (Tone 1) и тон 2 (Tone 2).

Тон 1 генерируется путем передачи с частотой 30 МГц по очереди кодов, состоящих из 16 единиц, и кодов, состоящих из 16 нулей. Результирующий сигнал имеет частоту примерно 0.9375 МГц.

Тон 2 генерируется путем передачи с частотой 30 МГц по очереди кодов, состоящих из 8 единиц, и кодов, состоящих из 8 нулей. Результирующий сигнал имеет частоту примерно 1.875 МГц.

Взаимодействие между концентратором и узлом происходит путем параллельной передачи по двум парам комбинации из указанных двух тонов.

В следующей таблице приведены значения возможных 4-х комбинаций тонов.

Комбинация тонов	**Значение при приеме узлом**	**Значение при приеме концентратором**
1 - 1	Простой (Idle)	Простой (Idle)
1 - 2	Поступление кадра	Запрос на передачу кадра с нормальным приоритетом
2 -1	Зарезервировано	Запрос на передачу кадра с высоким приоритетом
2 - 2	Запрос на инициализацию процедуры подготовки линии	Запрос на инициализацию процедуры подготовки линии

Состояние простоя означает, что концентратор или узел не имеют кадров, ожидающих передачи.

Состояние "поступление кадра" означает, что на данный порт может быть передан кадр. Узел должен прекратить передачу сигнальных тонов по каналам 2 и 3 для того, чтобы быть готовым принять кадр.

13. ТЕХНОЛОГИЯ FAST EHTERNET

Технология Fast Ethernet (IEEE 802.3u) является эволюционным развитием классической технологии Ethernet. Ее основными достоинствами являются:

- увеличение пропускной способности сегментов сети до 100 Мбит/с;

- сохранение метода случайного доступа Ethernet;

- сохранение звездообразной топологии сетей и поддержка традиционных сред передачи данных - витой пары и оптоволоконного кабеля.

Указанные свойства позволяют осуществлять постепенный переход от сетей 10Base-T - наиболее популярного варианта Ethernet - к скоростным сетям, сохраняющим значительную преемственность с хорошо знакомой технологией: Fast Ethernet не требует коренного переобучения персонала и замены оборудования во всех узлах сети. Официальный стандарт 100Base-T (802.3u) установил три различных спецификации для физического уровня (в терминах семиуровневой модели OSI) для поддержки следующих типов кабельных систем:

- 100Base-TX для двухпарного кабеля на неэкранированной витой паре UTP Category 5, или экранированной витой паре STP Type 1;

- 100Base-T4 для четырехпарного кабеля на неэкранированной витой паре UTP Category 3, 4 или 5;

- 100Base-FX для многомодового оптоволоконного кабеля.

13.1. Создание стандарта Fast Ethernet

В 1992 году группа производителей сетевого оборудования, включая таких лидеров технологии Ethernet как SynOptics, 3Com и ряд других, образовали некоммерческое объединение *Fast Ethernet Alliance* для разработки стандарта на новую технологию, которая обобщила бы достижения отдельных компаний в области Ethernet-преемственного высокоскоростного стандарта. Новая технология получила название Fast Ethernet.

В мае 1995 года комитет IEEE принял спецификацию Fast Ethernet в качестве стандарта 802.3u, который не является самостоятельным стандартом, а

представляет собой дополнение к существующему стандарту 802.3 в виде глав с 21 по 30. Отличия Fast Ethernet от Ethernet сосредоточены на физическом уровне (рисунок 13.1).

Более сложная структура физического уровня технологии Fast Ethernet вызвана тем, что в ней используется три варианта кабельных систем - оптоволокно, 2-х парная витая пара категории 5 и 4-х парная витая пара категории 3, причем по сравнению с вариантами физической реализации Ethernet (а их насчитывается шесть), здесь отличия каждого варианта от других глубже - меняется и количество проводников, и методы кодирования. А так как физические варианты Fast Ethernet создавались одновременно, а не эволюционно, как для сетей Ethernet, то имелась возможность детально определить те подуровни физического уровня, которые не изменяются от варианта к варианту, и остальные подуровни, специфические для каждого варианта.

Подуровни LLC и MAC в стандарте Fast Ethernet не претерпели изменений. Форматы кадров технологии Fast Ethernet не отличаются от форматов кадров технологий 10-Мегабитного Ethernet'a. На рисунке 13.2 приведен формат MAC-кадра Ethernet, а также временные параметры его передачи по сети для скорости 10 Мбит/с и для скорости 100 Мбит/с.

В кадрах стандарта Ethernet-II (или Ethernet DIX), опубликованного компаниями Xerox, Intel и Digital еще до появления стандарта IEEE 802.3, вместо двухбайтового поля L (длина поля данных) используется двухбайтовое поле T (тип кадра). Значение поля типа кадра всегда больше 1518 байт, что позволяет легко различить эти два разных формата кадров Ethernet DIX и IEEE 802.3.

Все времена передачи кадров Fast Ethernet в 10 раз меньше соответствующих времен технологии 10-Мегабитного Ethernet: битовый интервал составляет 10 нс вместо 100 нс, а межкадровый интервал - 0.96 мкс вместо 9.6 мкс соответственно.

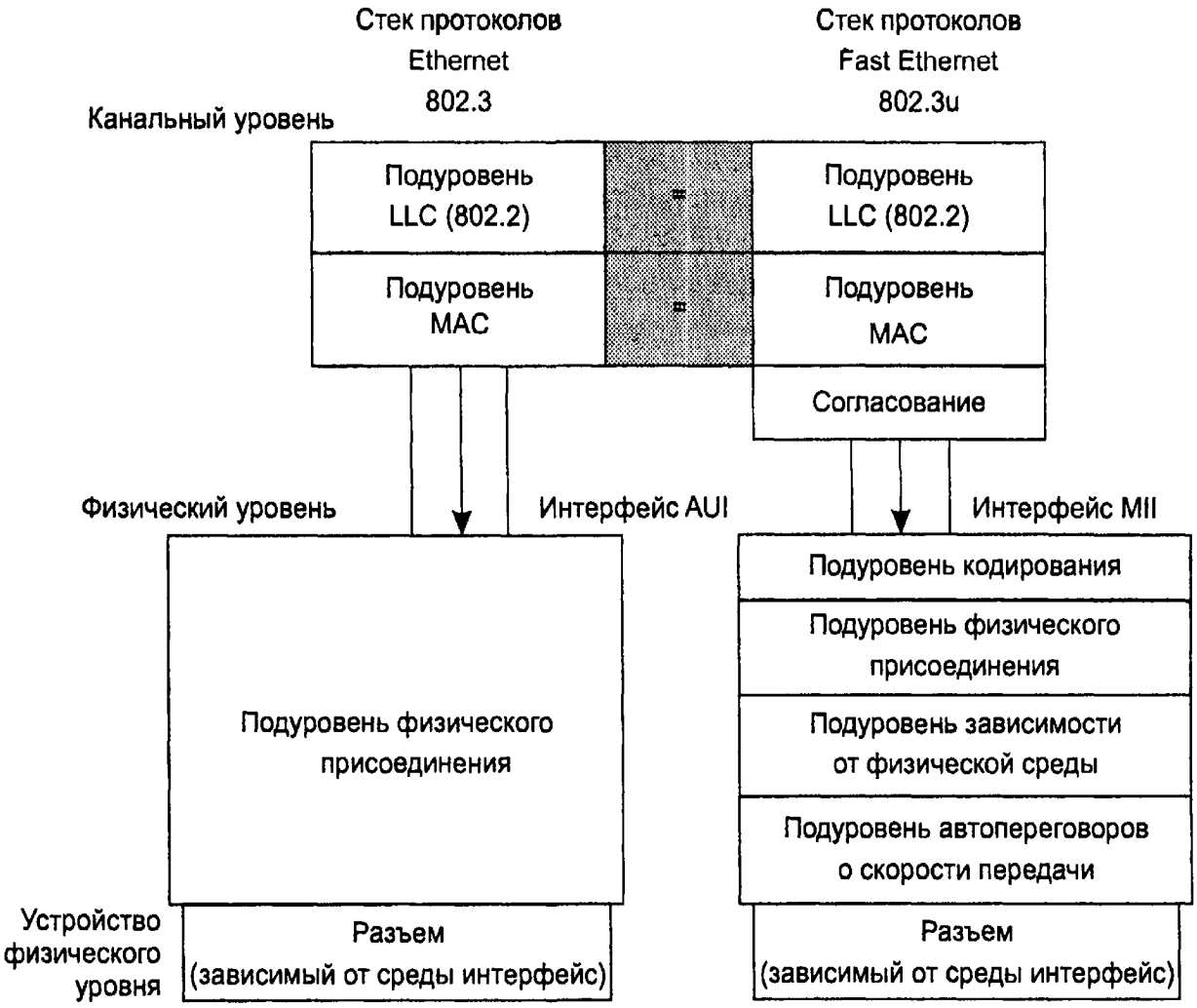


Рисунок 15.1. Отличия стеков протоколов 100Base-T и 10Base-T

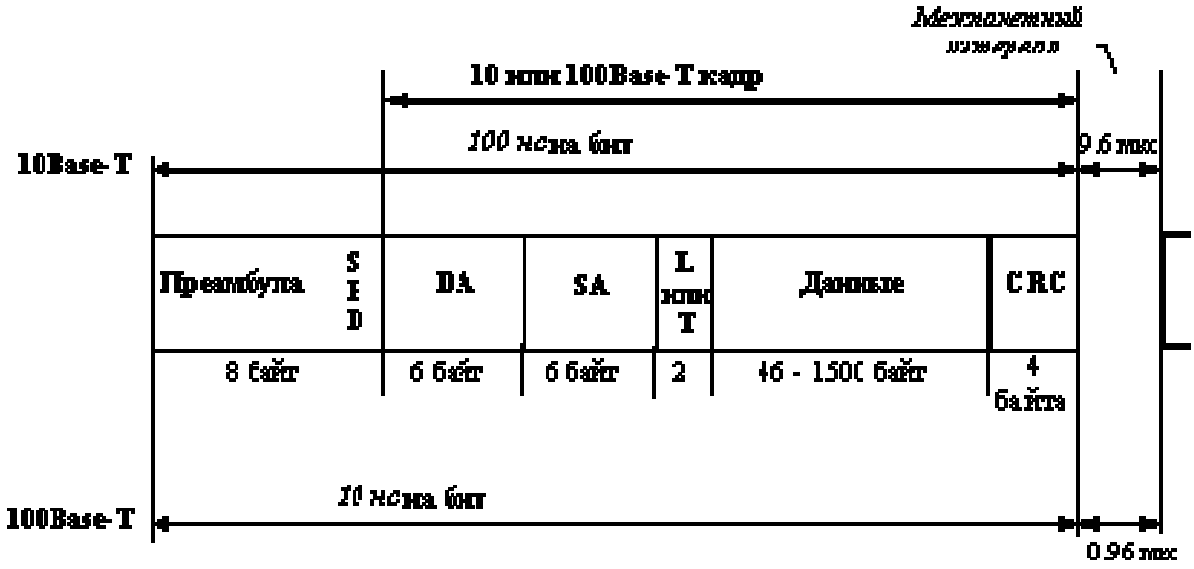


Рисунок 15.2. Формат MAC-кадра и времена его передачи

13.2. Структура физического уровня и его связь с MAC-подуровнем

Для технологии Fast Ethernet разработаны различные варианты физического уровня, отличающиеся не только типом кабеля и электрическими параметрами импульсов, как это сделано в технологии 10 Мбит/с Ethernet, но и способом кодирования сигналов и количеством используемых в кабеле проводников. Поэтому физический уровень Fast Ethernet имеет более сложную структуру, чем классический Ethernet. Эта структура представлена на рис.13.3.

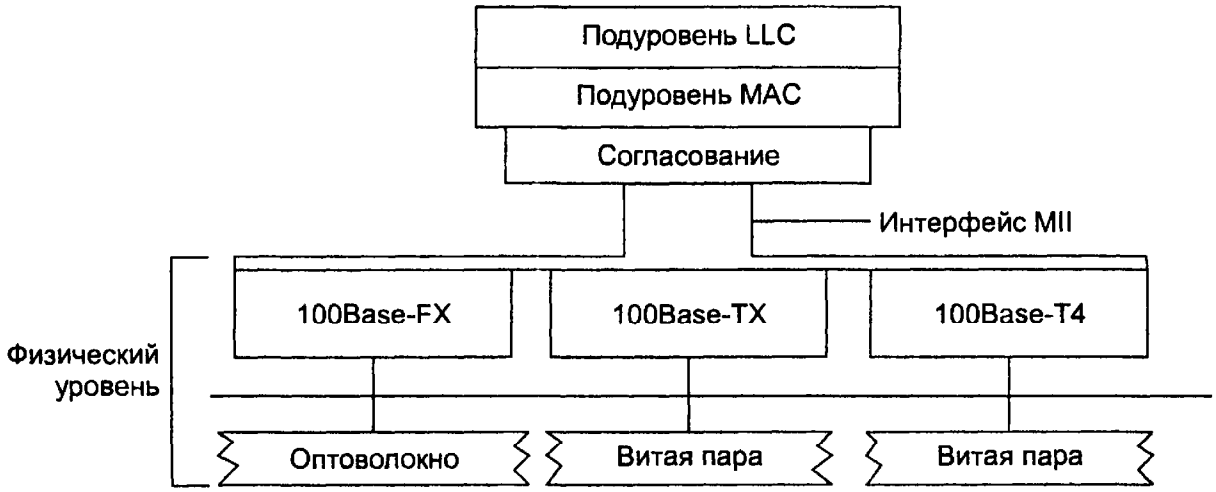


Рис. 13.2. Структура физического уровня Fast Ethernet

Рисунок 13.3. Структура физического уровня Fast Ethernet

Физический уровень состоит из трех подуровней:

- Уровень согласования (reconciliation sublayer);
- Независимый от среды интерфейс (Media Independent Interface, MII);
- Устройство физического уровня (Physical layer device, PHY).

Устройство физического уровня (PHY) обеспечивает кодирование данных, поступающих от MAC-подуровня для передачи их по кабелю определенного типа, синхронизацию передаваемых по кабелю данных, а также прием и декодирование данных в узле-приемнике.

Интерфейс MII поддерживает независимый от используемой физической среды способ обмена данными между MAC-подуровнем и подуровнем PHY. Этот интерфейс аналогичен по назначению интерфейсу AUI классического Ethernet'а за

исключением того, что интерфейс AUI располагался между подуровнем физического кодирования сигнала (для любых вариантов кабеля использовался одинаковый метод физического кодирования - манчестерский код) и подуровнем физического присоединения к среде, а интерфейс MII располагается между MAC-подуровнем и подуровнями кодирования сигнала, которых в стандарте Fast Ethernet три - FX, TX и T4.

Интерфейс MII

MII использует 4-битные порции данных для параллельной передачи их между MAC и PHY. Канал передачи данных от MAC к PHY образован 4-битной шиной данных, которая синхронизируется тактовым сигналом, генерируемым PHY, а также сигналом "Передача", генерируемым MAC-подуровнем.

Аналогично, канал передачи данных от PHY к MAC образован другой 4-битной шиной данных, которая синхронизируется тактовым сигналом и сигналом "Прием", которые генерируются PHY.

Если устройство PHY обнаружило ошибку в состоянии физической среды, то оно может передать сообщение об этом на подуровень MAC в виде сигнала "Ошибка приема" (receive error). MAC-подуровень (или повторитель) сообщают об ошибке устройству PHY с помощью сигнала "Ошибка передачи" (transmit error).

В MII определена двухпроводная шина для обмена между MAC и PHY управляющей информацией. MAC-подуровень использует эту шину для передачи PHY данных о режиме его работы. PHY передает по этой шине информацию по запросу о статусе порта и линии. Данные о конфигурации, а также о состоянии порта и линии хранятся соответственно в двух регистрах: регистре управления (Control Register) и регистре статуса (Status Register).

Регистр управления используется для установки скорости работы порта, для указания, будет ли порт принимать участие в процессе автопереговоров о скорости линии, для задания режима работы порта - полудуплексный или полнодуплексный, и т.п. Функция автопереговоров (Auto-negotiation) позволяет двум устройствам, соединенным одной линией связи, автоматически, без

вмешательства оператора, выбрать наиболее высокоскоростной режим работы, который будет поддерживается обоими устройствами.

Регистр статуса содержит информацию о действительном текущем режиме работы порта, в том числе и в том случае, когда режим выбран в результате проведения автопереговоров.

Регистр статуса может содержать данные об одном из следующих режимов:

- 100Base-T4;
- 100Base-TX full-duplex;
- 100Base-TX half-duplex;
- 10 Mb/s full-duplex;
- 10Mb/s half-duplex;
- Ошибка на дальнем конце линии.

13.3. Физический уровень 100Base-FX - многомодовое оптоволокно

Физический уровень PHY ответственен за прием данных в параллельной форме от MAC-подуровня, трансляцию их в один (TX или FX) или три последовательных потока бит с возможностью побитной синхронизации и передачу их через разъем на кабель. Аналогично, на приемном узле уровень PHY должен принимать сигналы по кабелю, определять моменты синхронизации бит, извлекать биты из физических сигналов, преобразовывать их в параллельную форму и передавать подуровню MAC.

Структура физического уровня 100Base-FX представлена на рисунке 13.4.

Эта спецификация определяет работу протокола Fast Ethernet по многомодовому оптоволокну в полудуплексном и полнодуплексном режимах на основе хорошо проверенной схемы кодирования и передачи оптических сигналов, использующейся уже на протяжении ряда лет в стандарте FDDI. Как и в стандарте FDDI, каждый узел соединяется с сетью двумя оптическими волокнами, идущими от приемника (R_x) и от передатчика (T_x).

Между спецификациями PHY FX и PHY TX есть много общего, поэтому общие для двух спецификаций свойства будут даваться под обобщенным названием PHY FX/TX.

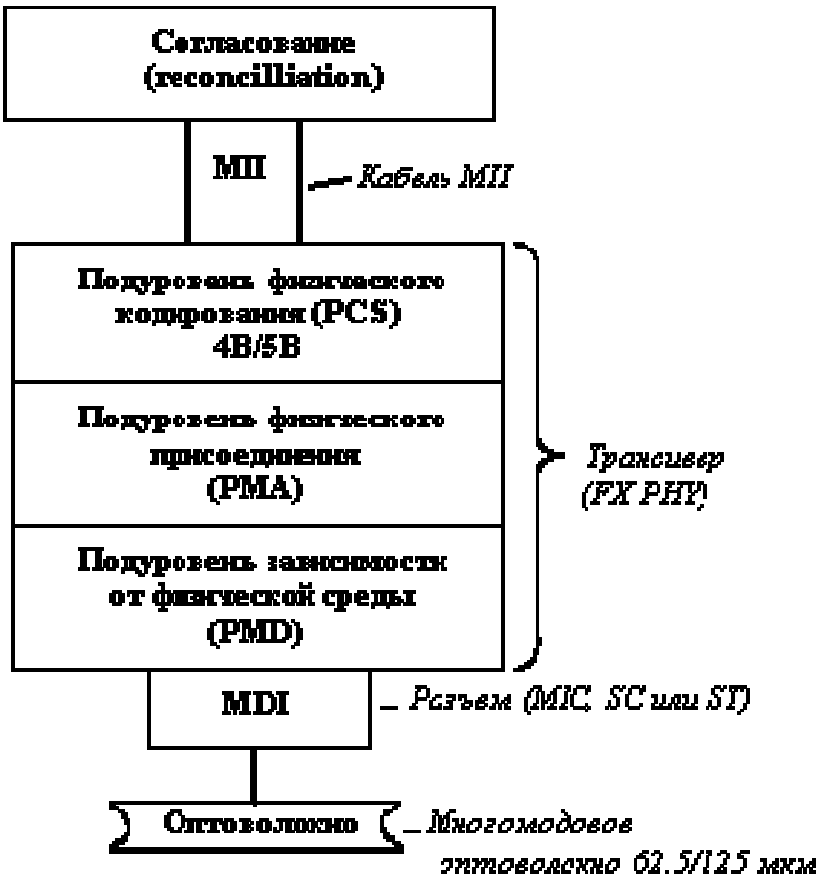


Рисунок 13.4. Физический уровень PHY FX

Метод кодирования 4B/5B определен в стандарте FDDI, и он без изменений перенесен в спецификацию PHY FX/TX. При этом методе каждые 4 бита данных MAC-подуровня (называемых символами) представляются 5 битами. Использование избыточного бита позволяет применить потенциальные коды при представлении каждого из пяти бит в виде электрических или оптических импульсов. Потенциальные коды обладают по сравнению с манчестерскими кодами более узкой полосой спектра сигнала, а, следовательно, предъявляют меньшие требования к полосе пропускания кабеля. Однако, прямое использование потенциальных кодов для передачи исходных данных без избыточного бита невозможно из-за плохой самосинхронизации приемника и

источника данных: при передаче длинной последовательности единиц или нулей в течение долгого времени сигнал не изменяется, и приемник не может определить момент чтения очередного бита.

При использовании пяти бит для кодирования шестнадцати исходных 4-х битовых комбинаций, можно построить такую таблицу кодирования, в которой любой исходный 4-х битовый код представляется 5-ти битовым кодом с чередующимися нулями и единицами. Тем самым обеспечивается синхронизация приемника с передатчиком. Так как исходные биты MAC-подуровня должны передаваться со скоростью 100Мбит/с, то наличие одного избыточного бита вынуждает передавать биты результирующего кода 4B/5B со скоростью 125 Мбит/с, то есть межбитовое расстояние в устройстве PHY составляет 8 наносекунд.

Так как из 32 возможных комбинаций 5-битовых порций для кодирования порций исходных данных нужно только 16, то остальные 16 комбинаций в коде 4B/5B используются в служебных целях.

Наличие служебных символов позволило использовать в спецификациях FX/TX схему непрерывного обмена сигналами между передатчиком и приемником и при свободном состоянии среды, что отличает их от спецификации 10Base-T, когда незанятое состояние среды обозначается полным отсутствием на ней импульсов информации. Для обозначения незанятого состояния среды используется служебный символ Idle (11111), который постоянно циркулирует между передатчиком и приемником, поддерживая их синхронизм и в периодах между передачами информации, а также позволяя контролировать физическое состояние линии. Существование запрещенных комбинаций символов позволяет отбраковывать ошибочные символы, что повышает устойчивость работы сетей с PHY FX/TX.

Для отделения кадра Ethernet от символов Idle используется комбинация символов Start Delimiter (пара символов JK), а после завершения кадра перед первым символом Idle вставляется символ T (рисунок 13.6).

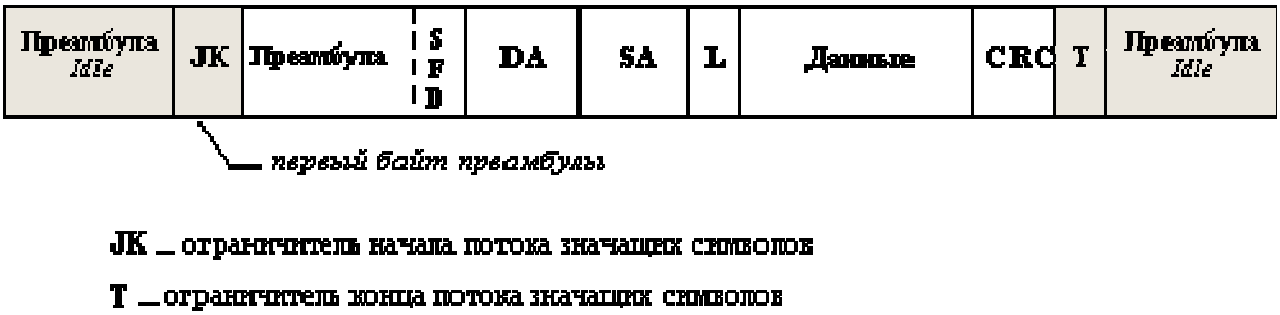


Рисунок 13.6. Непрерывный поток данных спецификаций PHY FX/TX

После преобразования 4-битовых порций MAC-кодов в 5-битовые порции PHY их необходимо представить в виде оптических или электрических сигналов в кабеле, соединяющем узлы сети. Спецификации PHY FX и PHY TX используют для этого различные методы физического кодирования - NRZI и MLT-3 соответственно. Эти же методы определены в стандарте FDDI для передачи сигналов по оптоволокну (спецификация PMD) и витой паре (спецификация TP-PMD).

Рассмотрим метод *NRZI - Non Return to Zero Invert to ones* - метод без возврата к нулю с инвертированием для единиц. Этот метод представляет собой модификацию простого потенциального метода кодирования, называемого *Non Return to Zero (NRZ),* когда для представления 1 и 0 используются потенциалы двух уровней. В методе NRZI также используется два уровня потенциала сигнала, но потенциал, используемый для кодирования текущего бита зависит от потенциала, который использовался для кодирования предыдущего бита (так называемое дифференциальное кодирование). Если текущий бит имеет значение 1, то текущий потенциал представляет собой инверсию потенциала предыдущего бита, независимо от его значения. Если же текущий бит имеет значение 0, то текущий потенциал повторяет предыдущий.

Основное преимущество NRZI кодирования по сравнению с NRZ кодированием в более надежном распознавании передаваемых 1 и 0 на линии в условиях помех.

13.4. Физический уровень 100Base-TX - двухпарная витая пара

Структура физического уровня спецификации PHY TX представлена на рисунке 13.10.

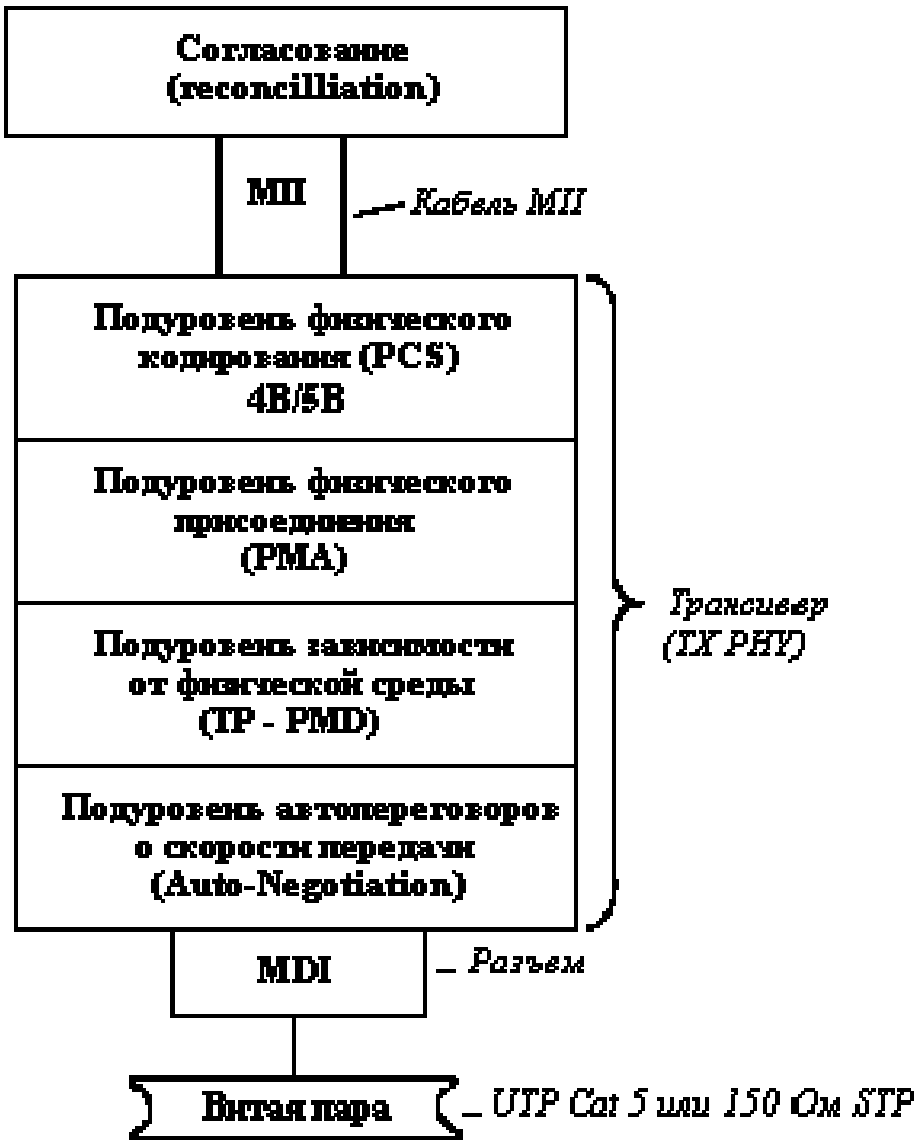


Рисунок 13.7. Структура физического уровня PHY TX

Основные отличия от спецификации PHY FX - использование метода MLT-3 для передачи сигналов 5-битовых порций кода 4B/5B по витой паре, а также наличие функции автопереговоров (Auto-negotiation) для выбора режима работы порта.

Метод MLT-3 использует потенциальные сигналы двух полярностей для представления 5-битовых порций информации (рисунок 13.8).

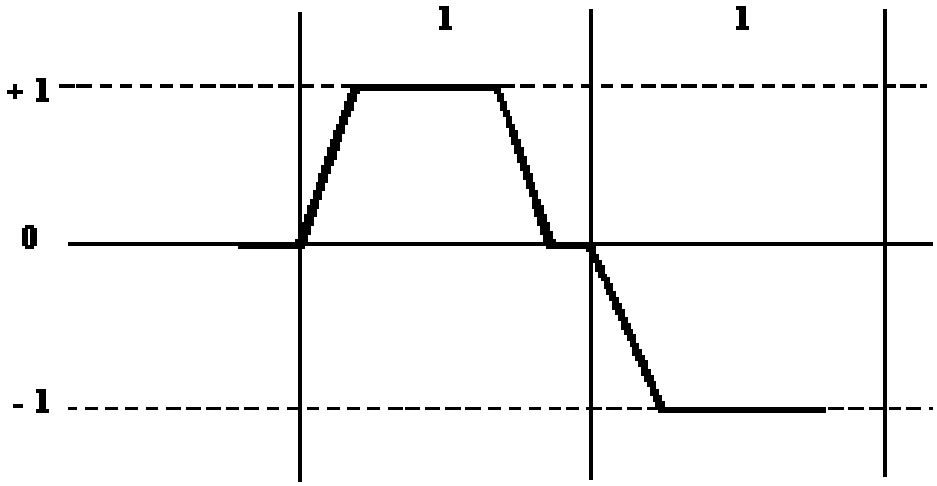


Рисунок 13.8. Метод кодирования MLT-3

Кроме использования метода MLT-3, спецификация PHY TX отличается от спецификации PHY FX тем, что в ней используется пара шифратор-дешифратор (scrambler/descrambler), как это определено в спецификации ANSI TP-PMD. Шифратор принимает 5-битовые порции данных от подуровня PCS, выполняющего кодирование 4B/5B, и зашифровывает сигналы перед передачей на подуровень MLT-3 таким образом, чтобы равномерно распределить энергию сигнала по всему частотному спектру - это уменьшает электромагнитное излучение кабеля.

Спецификации PHY TX и PHY T4 поддерживают функцию Auto-negotiation, с помощью которой два взаимодействующих устройства PHY могут автоматически выбрать наиболее эффективный режим работы.

Всего в настоящее время определено 5 различных режимов работы, которые могут поддерживать устройства PHY TX или PHY T4 на витых парах:

- 10Base-T (2 пары категории 3)
- 10Base-T full-duplex (2 пары категории 3)
- 100Base-TX (2 пары категории 5 (или Type 1 STP)
- 100Base-TX full-duplex (2 пары категории 5 (или Type 1 STP)
- 100Base-T4 (4 пары категории 3)

Режим 10Base-T имеет самый низкий приоритет при переговорном процессе, а режим 100Base-T4 - самый высокий. Переговорный процесс

происходит при включении питания устройства, а также может быть инициирован и в любой момент модулем управления.

Узлы, поддерживающие функцию Auto-negotiation, также используют существующую технологию сигналов проверки целостности линии, при этом они посылают пачки таких импульсов, инкапсулирующие информацию переговорного процесса Auto-negotiation. Такие пачки носят название *Fast Link Pulse burst (FLP).*

Устройство, начавшее процесс auto-negotiation, посылает своему партнеру пачку импульсов FLP, в котором содержится 8-битное слово, кодирующее предлагаемый режим взаимодействия, начиная с самого приоритетного, поддерживаемого данным узлом.

Если узел-партнер поддерживает функцию Auto-negotuiation и также может поддерживать предложенный режим, то он отвечает пачкой импульсов FLP, в которой подтверждает данный режим и на этом переговоры заканчиваются. Если же узел-партнер может поддерживать менее приоритетный режим, то он указывает его в ответе и этот режим выбирается в качестве рабочего. Таким образом, всегда выбирается наиболее приоритетный общий режим узлов.

Узел, который поддерживает только технологию 10Base-T, каждые 16 миллисекунд посылает импульсы для проверки целостности линии, связывающей его с соседним узлом. Такой узел не понимает запрос FLP, который делает ему узел с функцией Auto-negotiation, и продолжает посылать свои импульсы. Узел, получивший в ответ на запрос FLP только импульсы проверки целостности линии, понимает, что его партнер может работать только по стандарту 10Base-T и устанавливает этот режим работы и для себя.

13.5.Физический уровень 100Base-T4 - четырехпарная витая пара

Спецификация PHY T4 была разработана для того, чтобы можно было использовать для высокоскоростного Ethernet'а имеющуюся проводку на витой паре категории 3. Эта спецификация использует все 4 пары кабеля для того, чтобы можно было повысить общую пропускную способность за счет одновременной передачи потоков бит по нескольким витым парам.

Структура физического уровня PHY T4 изображена на рисунке 13.9.

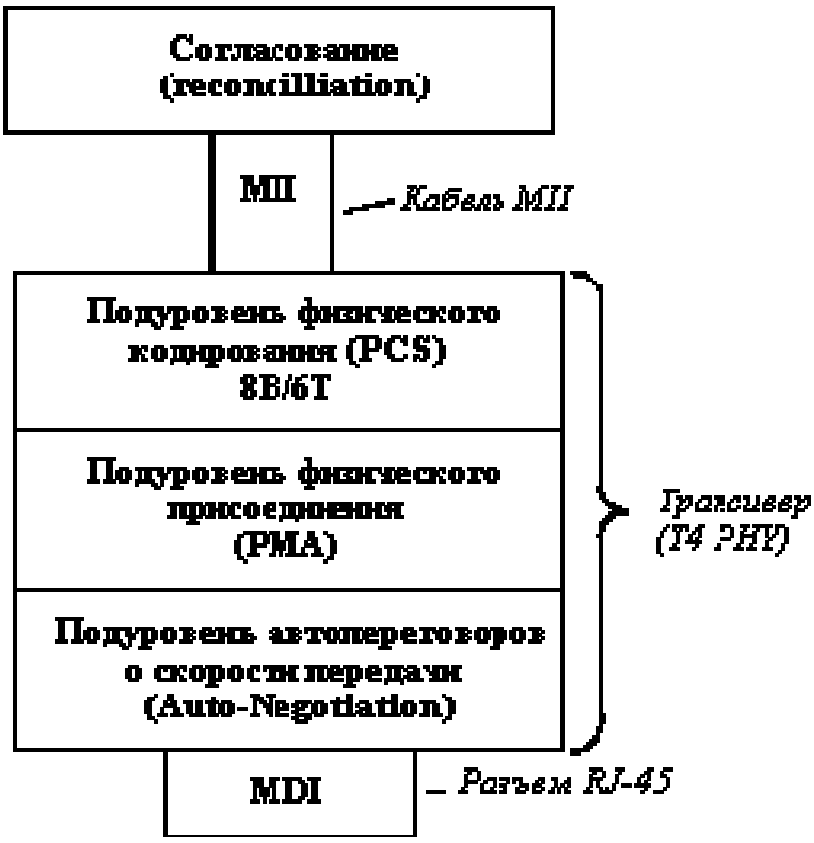


Рисунок 13.9. Физический уровень PHY T4

Вместо кодирования 4B/5B в этом методе используется кодирование 8B/6T. Каждые 8 бит информации MAC-уровня кодируются 6-ю троичными цифрами (ternary symbols), то есть цифрами, имеющими три состояния. Каждая троичная цифра имеет длительность 40 наносекунд. Группа из 6-ти троичных цифр затем передается на одну из трех передающих витых пар, независимо и последовательно. Четвертая пара всегда используется для прослушивания несущей частоты в целях обнаружения коллизии. Скорость передачи данных по каждой из трех передающих пар равна 33.3 Мбит/с, поэтому общая скорость протокола 100Base-T4 составляет 100 Мбит/с. В то же время из-за принятого способа кодирования скорость изменения сигнала на каждой паре равна всего 25 Мбод, что и позволяет использовать витую пару категории 3.

На рисунке 13.10 показано соединение порта MDI сетевого адаптера 100Base-T4 с портом MDI-X повторителя. Из рисунка видно, пара 1-2 всегда используется для передачи данных от порта MDI к порту MDI-X, пара 3-6 всегда

используется для приема данных портом MDI от порта MDI-X, а пары 4-5 и 7-8 являются двунаправленными и используются и для приема, и для передачи, в зависимости от потребности.

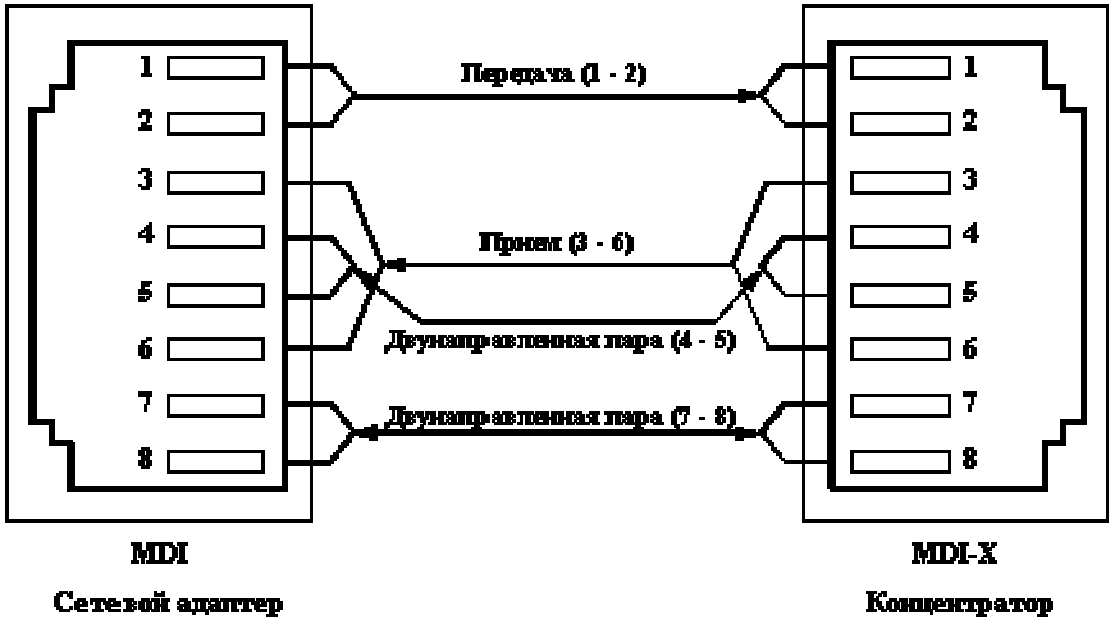


Рис. 13.10. Соединение узлов по спецификации PHY T4

13.6. Правила построения сегментов Fast Ethernet при использовании повторителей класса I и класса II

Технология Fast Ethernet, как и все некоаксиальные варианты Ethernet'a рассчитана на подключение конечных узлов - компьютеров с соответствующими сетевыми адаптерами - к многопортовым концентраторам-повторителям или коммутаторам.

Правила корректного построения сегментов сетей Fast Ethernet включают:

- ограничения на максимальные длины сегментов, соединяющих DTE с DTE;
- ограничения на максимальные длины сегментов, соединяющих DTE с портом повторителя;
- ограничения на максимальный диаметр сети;
- ограничения на максимальное число повторителей и максимальную длину сегмента, соединяющего повторители.

Ограничения длин сегментов DTE-DTE

В качестве *DTE (Data Terminal Equipment)* может выступать любой источник кадров данных для сети: сетевой адаптер, порт моста, порт маршрутизатора, модуль управления сетью и другие подобные устройства. Порт повторителя не является DTE. В типичной конфигурации сети Fast Ethernet несколько DTE подключается к портам повторителя, образуя сеть звездообразной топологии.

Спецификация IEEE 802.3u определяет следующие максимальные значения сегментов DTE-DTE:

Стандарт	Тип кабеля	Максимальная длина сегмента
100Base-TX	Category 5 UTP	100 метров
100Base-FX	многомодовое оптоволокно 62.5/125 мкм	412 метров (полудуплекс) 2 км (полный дуплекс)
100Base-T4	Category 3,4 или 5 UTP	100 метров

Ограничения, связанные с соединениями с повторителями

Повторители Fast Ethernet делятся на два класса.

Повторители класса I поддерживают все типы систем кодирования физического уровня: 100Base-TX/FX и 100Base-T4.

Повторители класса II поддерживают только один тип системы кодирования физического уровня - 100Base-TX/FX или 100Base-T4.

В одном домене коллизий допускается наличие только одного повторителя класса I. Это связано с тем, что такой повторитель вносит большую задержку при распространении сигналов из-за необходимости трансляции различных систем сигнализации.

Максимальное число повторителей класса II в домене коллизий - 2, причем они должны быть соединены между собой кабелем не длиннее 5 метров.

Небольшое количество повторителей Fast Ethernet не является серьезным препятствием при построении сетей. Во-первых, наличие стековых повторителей снимает проблемы ограниченного числа портов - все каскадируемые повторители

представляют собой один повторитель с достаточным числом портов - до нескольких сотен. Во-вторых, применение коммутаторов и маршрутизаторов делит сеть на несколько доменов коллизий, в каждом из которых обычно имеется не очень большое число станций.

В следующей таблице сведены правила построения сети на основе повторителе класса I.

Тип кабелей	**Максимальный диаметр сети** **Максимальная длина сегмента**		
Только витая пара (TX)	200 м	100 м	
Только оптоволокно (FX)	272 м	136 м	
Несколько сегментов на витой паре и один на оптоволокне	260 м	100 м (TX)	160 м (FX)
Несколько сегментов на витой паре и несколько сегментов на оптоволокне	272 м	100 м (TX)	136 м (FX)

Эти ограничения проиллюстрированы типовыми конфигурациями сетей, показанными на рисунке 15.14.

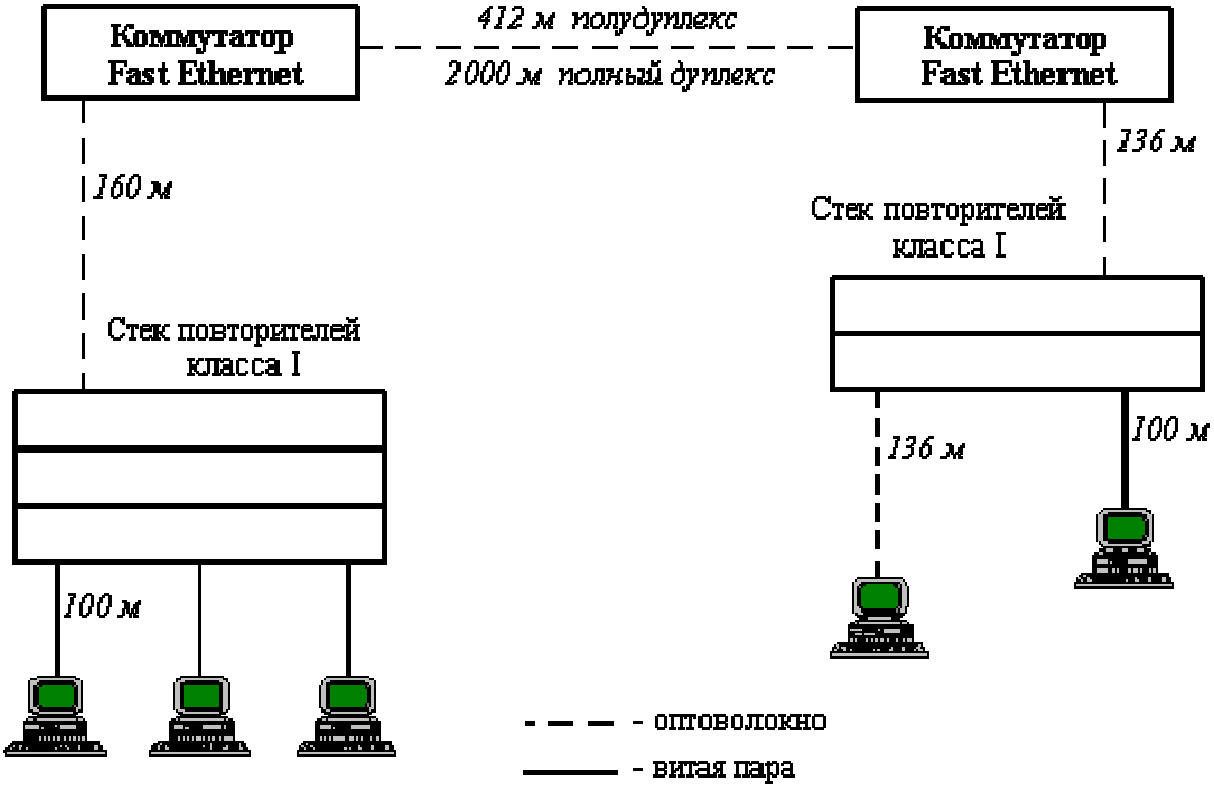


Рисунок 13.11. Примеры построения сети с помощью повторителей класса I

14. ТЕХНОЛОГИЯ GIGABITE EHTERNET

14.1. Хронология разработки стандарта

В марте 1996 года комитет IEEE 802.3 одобряет проект стандартизации Gigabit Ethernet 802.3z. В мае 1996 года 11 компаний (3Com Corp., Bay Networks Inc., Cisco Systems Inc., Compaq Computer Corp., Granite Systems Inc., Intel Corporation, LSI Logic, Packet Engines Inc., Sun Microsystems Computer Company, UB Networks и VLSI Technology) организовывают Gigabit Ethernet Alliance.

Альянс, объединяя усилия большого числа ведущих производителей сетевого оборудования на пути выработки единого стандарта и выпуска взаимосовместимых продуктов Gigabit Ethernet, преследует следующие цели:

- поддержка расширения технологий Ethernet и Fast Ethernet в ответ на потребность в более высокой скорости передачи;

- разработка технических предложений с целью включения в стандарт;

- выработка процедур и методов тестирования продуктов от различных поставщиков.

К началу 1998 года Альянс насчитывает уже более 100 компаний. Через Альянс обеспечивается обратная связь между техническим комитетом по стандартизации IEEE 802.3 и индустриальными производителями сетевого оборудования. Альянс увеличивает эффективность работы комитета и способствует более быстрому одобрению спецификаций стандартов Gigabit Ethernet IEEE 802.3z и IEEE 802.3ab. Наибольшие трудности вызывает физический уровень, а именно адаптация многомодового волокна и витой пары.

29 июня 1998 г. с задержкой примерно на полгода от первоначально запланированного графика, вызванной доработкой стандарта по отношению к использованию многомодового волокна (аномалия, получившая название DMD), принимается стандарт IEEE 802.3z (был одобрен в качестве стандарта пятый драфт 802.3z/D5). Соответствующие спецификации регламентируют использование одномодового, многомодового волокна, а также витой пары UTP cat.5 на короткие расстояния (до 25 м).

Стандартизация системы передачи Gigabit Ethernet по неэкранированной витой паре на расстояния до 100 м требовала разработки специального помехоустойчивого кода, для чего создается отдельный подкомитет P802.3ab. 28 июня 1999 г. принимается соответствующий стандарт (единогласно одобряется шестой драфт 802.3ab/D6).

Основная идея разработчиков стандарта Gigabit Ethernet состоит в максимальном сохранении идей классической технологии Ethernet при достижении битовой скорости в 1000 Мбит/с.

Так как при разработке новой технологии естественно ожидать некоторых технических новинок, идущих в общем русле развития сетевых технологий, то важно отметить, что Gigabit Ethernet, так же как и его менее скоростные собратья, на уровне протокола ***не будет*** поддерживать:

- качество обслуживания;
- избыточные связи;
- тестирование работоспособности узлов и оборудования (в последнем случае - за исключением тестирования связи порт - порт, как это делается для Ethernet l0Base-T и l0Base-F и Fast Ethernet).

Главная идея разработчиков технологии Gigabit Ethernet состоит в том, что существует, и будет существовать весьма много сетей, в которых высокая скорость магистрали и возможность назначения пакетам приоритетов в коммутаторах будут вполне достаточны для обеспечения качества транспортного обслуживания всех клиентов сети. И только в тех редких случаях, когда и магистраль достаточно загружена, и требования к качеству обслуживания очень жесткие, нужно применять технологию ATM, которая действительно за счет высокой технической сложности дает гарантии качества обслуживания для всех основных видов трафика.

Избыточные связи и тестирование оборудования не будут поддерживаться технологией Gigabit Ethernet из-за того, что с этими задачами хорошо справляются протоколы более высоких уровней, например Spanning Tree, протоколы маршрутизации и т. п. Поэтому разработчики технологии решили, что

нижний уровень просто должен быстро передавать данные, а более сложные и более редко встречающиеся задачи (например, приоритезация трафика) должны передаваться верхним уровням.

Что же общего имеется в технологии Gigabit Ethernet по сравнению с технологиями Ethernet и Fast Ethernet?

- Сохраняются все форматы кадров Ethernet.
- По-прежнему будут существовать полудуплексная версия протокола, поддерживающая метод доступа CSMA/CD, и полнодуплексная версия, работающая с коммутаторами. По поводу сохранения полудуплексной версии протокола сомнения были еще у разработчиков Fast Ethernet, так как сложно заставить работать алгоритм CSMA/CD на высоких скоростях. Однако метод доступа остался неизменным в технологии Fast Ethernet, и его решили оставить в новой технологии Gigabit Ethernet. Сохранение недорогого решения для разделяемых сред позволит применить Gigabit Ethernet в небольших рабочих группах, имеющих быстрые серверы и рабочие станции.
- Поддерживаются все основные виды кабелей, используемых в Ethernet и Fast Ethernet: волоконно-оптический, витая пара категории 5, коаксиал.

14.2. Архитектура стандарта Gigabit Ethernet

На рис.14.1 показана структура уровней Gigabit Ethernet. Как и в стандарте Fast Ethernet, в Gigabit Ethernet не существует универсальной схемы кодирования сигнала, которая была бы идеальной для всех физических интерфейсов - так, с одной стороны, для стандартов 1000Base-LX/SX/CX используется кодирование 8B/10B, а с другой стороны, для стандарта 1000Base-T используется специальный расширенный линейный код TX/T2. Функцию кодирования выполняет подуровень кодирования PCS, размещенный ниже среданезависимого интерфейса GMII.

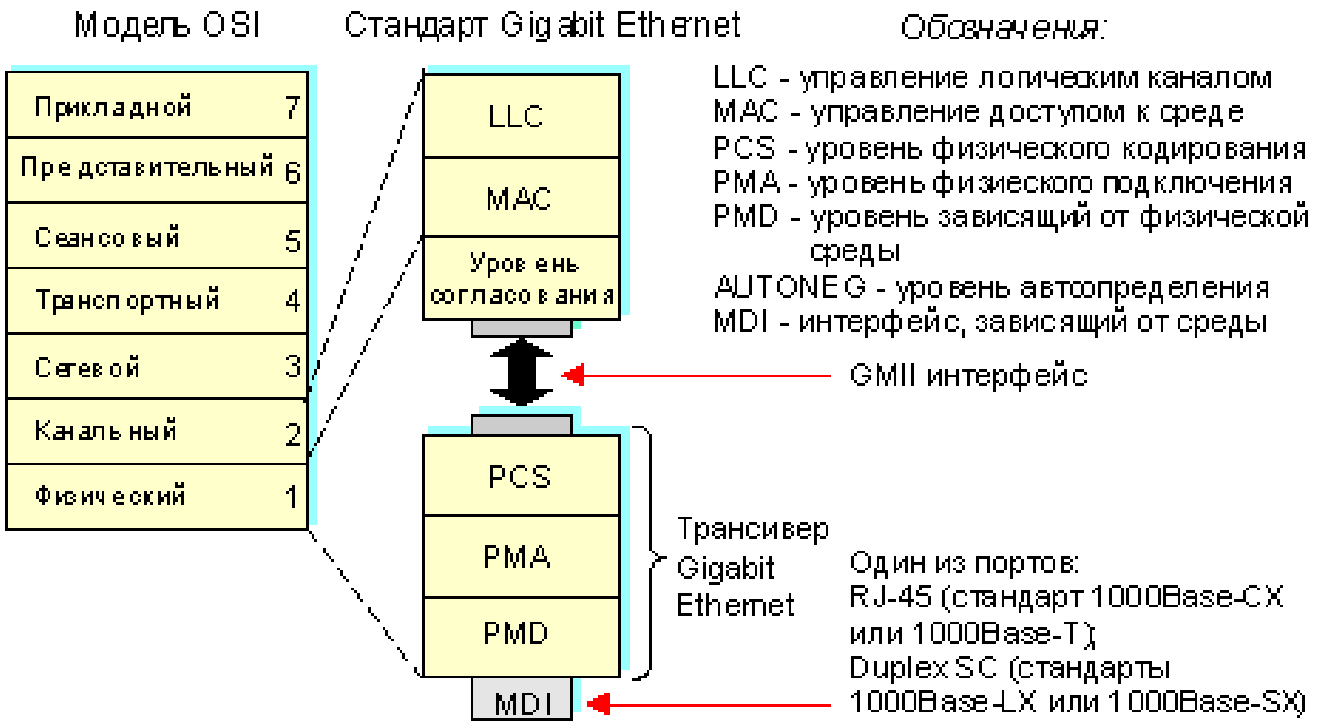


Рисунок 14.1. Структура уровней стандарта Gigabit Ethernet, GMII интерфейс и трансивер Gigabit Ethernet

GMII интерфейс. Среданезависимый интерфейс GMII (gigabit media independent interface) обеспечивает взаимодействие между уровнем MAC и физическим уровнем. GMII интерфейс является расширением интерфейса MII и может поддерживать скорости 10, 100 и 1000 Мбит/с. Он имеет отдельные 8-битные приемник и передатчик, и может поддерживать как полудуплексный, так и дуплексный режимы. Кроме этого, GMII интерфейс несет один сигнал, обеспечивающий синхронизацию (clock signal), и два сигнала состояния линии - первый (в состоянии ON) указывает наличие несущей, а второй (в состоянии ON) говорит об отсутствии коллизий - и еще несколько других сигнальных каналов и питание. Трансиверный модуль, охватывающий физический уровень и обеспечивающий один из физических средазависимых интерфейсов, может подключаться, например, к коммутатору Gigabit Ethernet посредством GMII интерфейса.

Подуровень физического кодирования PCS. При подключении интерфейсов группы 1000Base-X, подуровень PCS использует блочное

избыточное кодирование 8B10B, заимствованное из стандарта ANSI X3T11 Fibre Channel. Аналогично рассмотренному стандарту FDDI, только на основе более сложной кодовой таблицы каждые 8 входных битов, предназначенных для передачи на удаленный узел, преобразовываются в 10-битные символы (code groups). Кроме этого в выходном последовательном потоке присутствуют специальные контрольные 10-битные символы. Примером контрольных символов могут служить символы, используемые для расширения носителя (дополняют кадр Gigabit Ethernet до его минимально размера 512 байт). При подключении интерфейса 1000Base-T, подуровень PCS осуществляет специальное помехоустойчивое кодирование, для обеспечения передачи по витой паре UTP Cat.5 на расстояние до 100 метров - линейный код TX/T2, разработанный компанией Level One Communications.

Два сигнала состояния линии - сигнал наличие несущей и сигнал отсутствие коллизий - генерируются этим подуровнем.

Подуровни PMA и PMD. Физический уровень Gigabit Ethernet использует несколько интерфейсов, включая традиционную витую пару категории 5, а также многомодовое и одномодовое волокно. Подуровень PMA преобразует параллельный поток символов от PCS в последовательный поток, а также выполняет обратное преобразование (распараллеливание) входящего последовательного потока от PMD. Подуровень PMD определяет оптические/электрические характеристики физических сигналов для разных сред. Всего определяются 4 различный типа физических интерфейса среды, которые отражены в спецификация стандарта 802.3z (1000Base-X) и 802.3ab (1000Base-T), рис.14.2.

В стандарте 802.3z определены следующие типы физической среды:

- одномодовый волоконно-оптический кабель;
- многомодовый волоконно-оптический кабель 62,5/125;
- многомодовый волоконно-оптический кабель 50/125;
- двойной коаксиал с волновым сопротивлением 75 Ом.

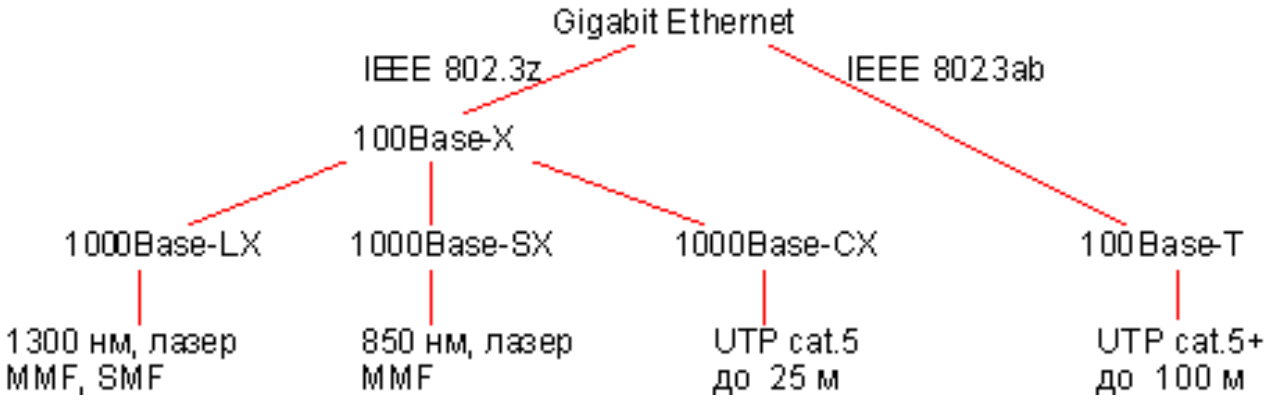


Рисунок 14.2.Физические интерфейсы стандарта Gigabit Ethernet

Многомодовый кабель. Для передачи данных по традиционному для компьютерных сетей многомодовому волоконно-оптическому кабелю стандарт определяет применение излучателей, работающих на двух длинах волн: 1300 и 850 нм. Применение светодиодов с длиной волны 850 нм объясняется тем, что они намного дешевле, чем светодиоды, работающие на волне 1300 нм, хотя при этом максимальная длина кабеля уменьшается, так как затухание многомодового оптоволокна на волне 850 м более чем в два раза выше, чем на волне 1300 нм. Однако возможность удешевления чрезвычайно важна для такой в целом дорогой технологии, как Gigabit Ethernet.

Для многомодового оптоволокна стандарт 802.3z определил спецификации 1000Base-SX и 1000Base-LX.

В первом случае используется длина волны 850 нм (S означает Short Wavelength, короткая волна), а во втором - 1300 нм (L - от Long Wavelength, длинная волна).

Для спецификации 1l000Base-SX предельная длина оптоволоконного сегмента для кабеля 62,5/125 оставляет 220 м, а для кабеля 50/125 - 500 м. Очевидно, что эти максимальные значения могут достигаться только для полнодуплексной передачи данных, так как время двойного оборота сигнала на двух отрезках 220 м равно 4400 bt, что превосходит предел 4095 bt даже без учета повторителя и сетевых адаптеров. Для полудуплексной передачи максимальные значения сегментов оптоволоконного кабеля всегда должны быть меньше 100 м.

Приведенные расстояния в 220 и 500 м рассчитаны для худшего по стандарту случая полосы пропускания многомодового кабеля, находящегося в пределах от 160 до 500 МГц/км. Реальные кабели обычно обладают значительно лучшими характеристиками, находящимися между 600 и 1000 МГц/км. В этом случае можно увеличить длину кабеля до примерно 800 м.

Одномодовый кабель. Для спецификации 1l000Base-LX в качестве источника излучения всегда применяется полупроводниковый лазер с длиной волны 1300 нм.

Основная область применения стандарта 1000Base-LX - это одномодовое оптоволокно. Максимальная длина кабеля для одномодового волокна равна 5000 м.

Спецификация 1000Base-LX может работать и на многомодовом кабеле. В этом случае предельное расстояние получается небольшим - 550 м. Это связано с особенностями распространения когерентного света в широком канале многомодового кабеля. Для присоединения лазерного трансивера к многомодовому кабелю необходимо использовать специальный адаптер.

Твинаксиальный кабель. В качестве среды передачи данных используется высококачественный твинаксиальный кабель (Twinax) с волновым сопротивлением 150 Ом (2x75 Ом). Данные посылаются одновременно по паре проводников, каждый из которых окружен экранирующей оплеткой. При этом получается режим полудуплексной передачи. Для обеспечения полнодуплексной передачи необходимы еще две пары коаксиальных проводников. Начал выпускаться специальный кабель, который содержит четыре коаксиальных проводника - так называемый Quad-кабель. Он внешне напоминает кабель категории 5 и имеет близкий к нему внешний диаметр и гибкость. Максимальная длина твинаксиального сегмента составляет всего 25 метров, поэтому это решение подходит для оборудования, расположенного в одной комнате.

14.3. Интерфейс 1000Base-X

Интерфейс 1000Base-X основывается на стандарте физического уровня Fibre Channel. Fibre Channel - это технология взаимодействия рабочих станций, суперкомпьютеров, устройств хранения и периферийных узлов. Fibre Channel имеет 4-х уровневую архитектуру. Два нижних уровня FC-0 (интерфейсы и среда) и FC-1 (кодирование/декодирование) перенесены в Gigabit Ethernet. Поскольку Fibre Channel является одобренной технологией, то такое перенесение сильно сократило время на разработку оригинального стандарта Gigabit Ethernet.

Блочный код 8B/10B аналогичен коду 4B/5B, принятому в стандарте FDDI. Однако код 4B/5B был отвергнут в Fibre Channel, потому что этот код не обеспечивает баланса по постоянному току. Отсутствие баланса потенциально может привести к зависящему от передаваемых данных нагреванию лазерных диодов, поскольку передатчик может передавать больше битов "1" (излучение есть), чем "0" (излучения нет), что может быть причиной дополнительных ошибок при высоких скоростях передачи.

1000Base-X подразделяется на три физических интерфейса, основные характеристики которых приведены ниже.

Интерфейс 1000Base-SX определяет лазеры с допустимой длиной излучения в пределах диапазона 770-860 нм, мощность излучения передатчика в пределах от -10 до 0 дБм, при отношении ON/OFF (сигнал / нет сигнала) не меньше 9 дБ. Чувствительность приемника -17 дБм, насыщение приемника 0 дБм;

Интерфейс 1000Base-LX определяет лазеры с допустимой длиной излучения в пределах диапазона 1270-1355 нм, мощность излучения передатчика в пределах от -13,5 до -3 дБм, при отношении ON/OFF (есть сигнал / нет сигнала) не меньше 9 дБ. Чувствительность приемника -19 дБм, насыщение приемника -3 дБм;

1000Base-CX экранированная витая пара (STP "twinax") на короткие расстояния.

14.4. Интерфейс 1000Base-T

1000Base-T - это стандартный интерфейс Gigabit Ethernet передачи по неэкранированной витой паре категории 5 и выше на расстояния до 100 метров. Для передачи используются все четыре пары медного кабеля, скорость передачи по одной паре 250 Мбит/с. Предполагается, что стандарт будет обеспечивать дуплексную передачу, причем данные по каждой паре будут передаваться одновременно сразу в двух направлениях - двойной дуплекс (dual duplex). 1000Base-T. Технически реализовать дуплексную передачу 1 Гбит/с по витой паре UTP cat.5 оказалось довольно сложно, значительно сложней, чем в стандарте 100Base-TX. Влияние ближних и дальних перекрестных помех от трех соседних витых пар на данную пару в четырехпарном кабеле требует разработки специальной скремблированной помехоустойчивой передачи, и интеллектуального узла распознавания и восстановления сигнала на приеме. Несколько методов кодирования первоначально рассматривались в качестве кандидатов на утверждение в стандарте 1000Base-T, среди которых: 5-уровневое импульсно-амплитудное кодирование PAM-5; квадратурная амплитудная модуляция QAM-25, и др. Ниже приведены кратко идеи PAM-5, окончательно утвержденного в качестве стандарта.

В пятиуровневом коде PAM 5 (рис.14.3) используется 5 уровней амплитуды и двухбитовое кодирование. Для каждой комбинации задается уровень напряжения. При двухбитовом кодировании для передачи информации необходимо четыре уровня (два во второй степени - 00, 01, 10, 11). Передача двух битов одновременно обеспечивает уменьшение в два раза частоты изменения сигнала. Пятый уровень добавлен для создания избыточности кода, используемого для исправления ошибок. Это дает дополнительный резерв соотношения сигнал / шум 6 дБ.

Код PAM5, использует 5 уровней потенциала: -2, -1,0, +1, +2. Поэтому за один такт по одной паре передается 2,322 бит информации. Следовательно, тактовую частоту вместо 250 МГц можно снизить до 125 МГц. При этом если использовать не все коды, а передавать 8 бит за такт (по 4 парам), то

выдерживается требуемая скорость передачи в 1000 Мбит/с и еще остается запас неиспользуемых кодов, так как код PAM5 содержит $5^4 = 625$ комбинаций, а если передавать за один такт по всем четырем парам 8 бит данных, то для этого требуется всего $2^8 = 256$ комбинаций. Оставшиеся комбинации приемник может использовать для контроля принимаемой информации и выделения правильных комбинаций на фоне шума. Код PAM5 на тактовой частоте 125 МГц укладывается в полосу 100 МГц кабеля категории 5.

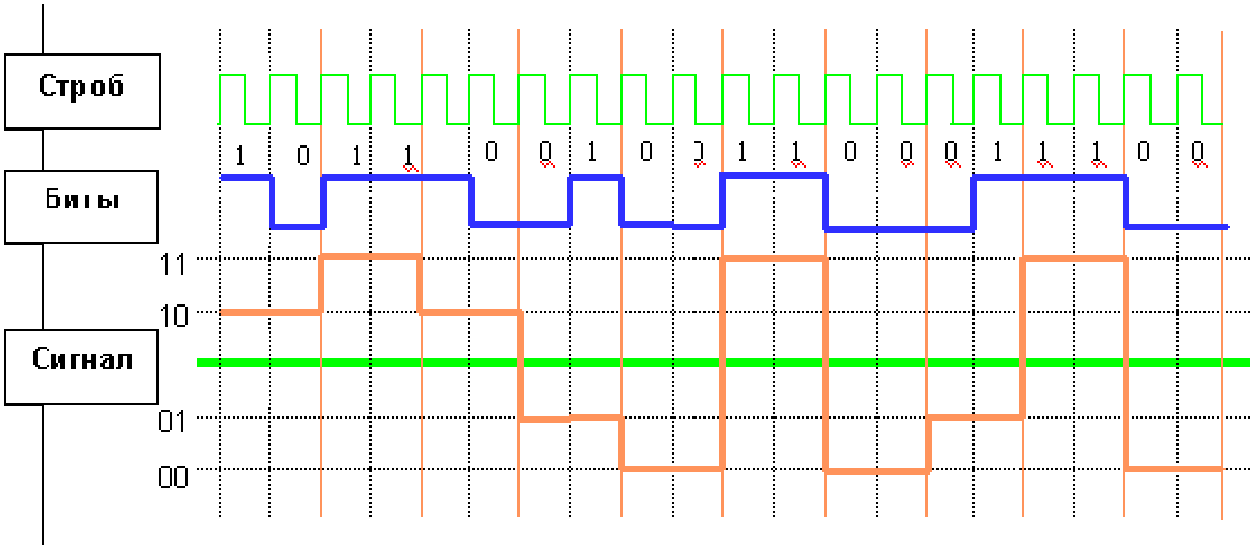


Рисунок 14.3. Код PAM 5

Для распознавания коллизий и организации полнодуплексного режима разработчики спецификации 802.3ab применили технику, используемую при организации дуплексного режима на одной паре проводов в современных модемах и аппаратуре передачи данных абонентских окончаний ISDN. Вместо передачи по разным парам проводов или разнесения сигналов двух одновременно работающих навстречу передатчиков по диапазону частот оба передатчика работают навстречу друг другу по каждой из 4-х пар в одном и том же диапазоне частот, так как используют один и тот же потенциальный код PAM5 (рис. 14.4). Схема гибридной развязки *H* позволяет приемнику и передатчику одного и того же узла использовать одновременно витую пару и для приема и для передачи (так же, как и в трансиверах коаксиального Ethernet).

Для отделения принимаемого сигнала от своего собственного приемник вычитает из результирующего сигнала известный ему свой сигнал. Естественно, что это не простая операция и для ее выполнения используются специальные цифровые сигнальные процессоры - DSP (Digital Signal Processor). Такая техника уже прошла проверку практикой, но в модемах и сетях ISDN она применялась совсем на других скоростях.

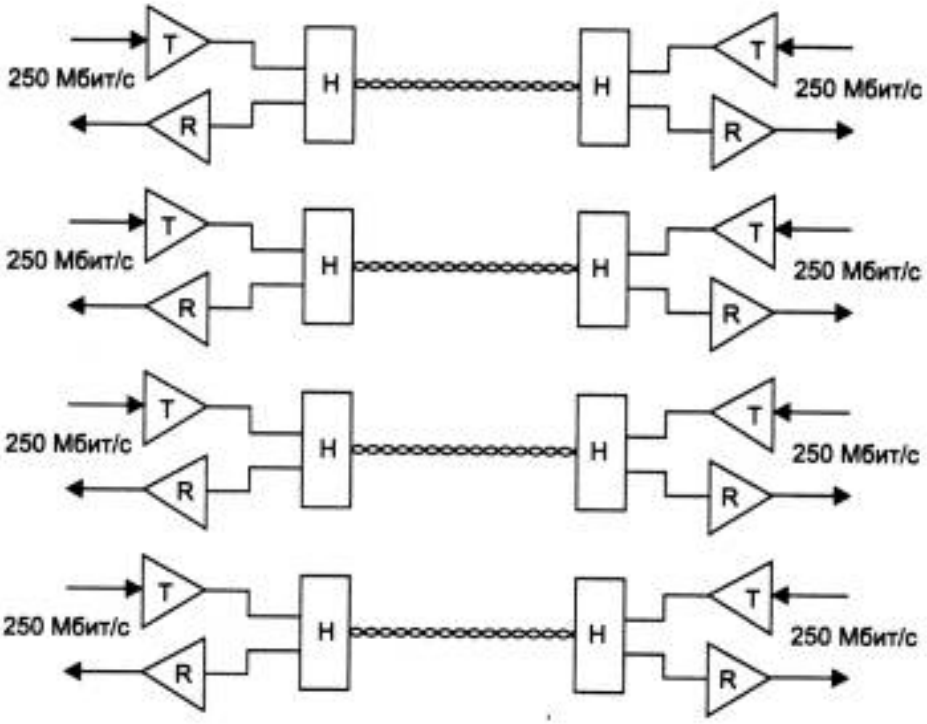


Рисунок 14.4. Двунаправленная передача по четырем парам UTP категории 5

При полудуплексном режиме работы получение встречного потока данных считается коллизией, а для полнодуплексного режима работы - нормальной ситуацией.

14.5. Уровень MAC

Уровень MAC стандарта Gigabit Ethernet использует тот же самый протокол передачи CSMA/CD что и его предки Ethernet и Fast Ethernet. Основные ограничения на максимальную длину сегмента (или коллизионного домена) определяются этим протоколом.

В стандарте Ethernet IEEE 802.3 принят минимальный размер кадра 64 байта. Именно значение минимального размера кадра определяет максимальное допустимое расстояние между станциями (диаметр коллизионного домена). Время, которого станция передает такой кадр - время канала - равно 512 bt или 51,2 мкс. Максимальная длина сети Ethernet определяется из условия разрешения коллизий, а именно время, за которое сигнал доходит до удаленного узла и возвращается обратно не должно превышать 512 bt (без учета преамбулы).

При переходе от Ethernet к Fast Ethernet скорость передачи возрастает, а время трансляции кадра длины 64 байта соответственно сокращается - оно равно 512 bt или 5,12 мкс (в Fast Ethernet 1 bt = 0,01 мкс). Для того, чтобы можно было обнаруживать все коллизии до конца передачи кадра, как и раньше необходимо удовлетворить одному из условий:

- сохранить прежнюю максимальную длину сегмента, но увеличить время канала (и следовательно увеличить минимальную длину кадра), или

- сохранить время канала, (сохранить прежний размер кадра), но уменьшить максимальную длину сегмента.

При разработке стандарта Gigabit Ethernet было принято решение увеличить время канала. В Gigabit Ethernet оно составляет 4096 bt и в 8 раз превосходит время канала Ethernet и Fast Ethernet. Но, чтобы поддержать совместимость со стандартами Ethernet и Fast Ethernet, минимальный размер кадра не был увеличен, а было добавлено к кадру дополнительное поле, получившее название "расширение носителя". Символы в дополнительном поле обычно не несут служебной информации, но они заполняют канал и увеличивают "коллизионное окно". В результате, коллизия будет регистрироваться всеми станциями при большем диаметре коллизионного домена.

Если станция желает передать короткий (меньше 512 байт) кадр, до при передаче добавляется это поле - расширение носителя, дополняющее кадр до 512 байт. Поле контрольной суммы вычисляется только для оригинального кадра и не распространяется на поле расширения. При приеме кадра поле расширения отбрасывается. Поэтому уровень LLC даже и не знает о наличии поля

расширения. Если размер кадра равен или превосходит 512 байт, то поле расширения носителя отсутствует. На рис 14.5 показан формат кадра Gigabit Ethernet при использовании расширения носителя.

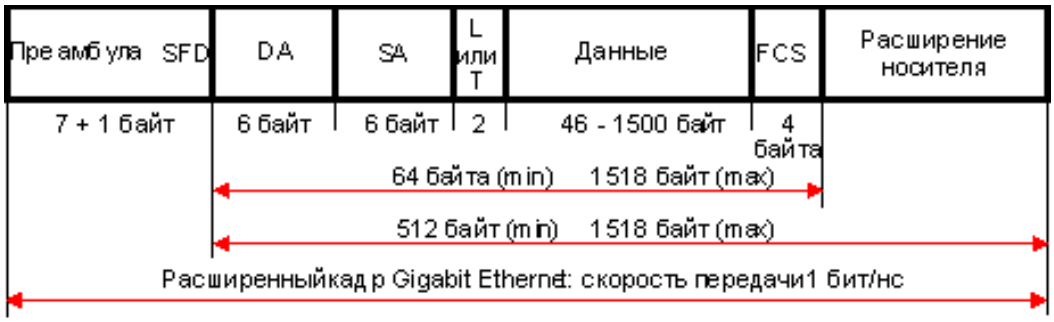


Рисунок 14.5. Кадр Gigabit Ethernet с полем расширения носителя

Пакетная перегруженность (Packet Bursting). Расширение носителя привело к излишней трате полосы пропускания. До 448 байт (512-64) может расходоваться в холостую при передаче короткого кадра. На стадии разработки стандарта Gigabit Ethernet компанией NBase Communications было внесено предложение по модернизации стандарта. Эта модернизация, получившая название **пакетная перегруженность**, позволяет эффективней использовать поле расширения. Если у станции/коммутатора имеется несколько небольших кадров для отправки, то первый кадр дополняется полем расширения носителя до 512 байт, и отправляется. Остальные кадры отправляются вслед с минимальным межкадровым интервалом в 96 bt, с одним важным исключением - межкадровый интервал заполняется символами расширения. Таким образом, среда не замолкает между посылками коротких оригинальных кадров, и ни какое другое устройство сети не может вклиниться в передачу. Такое пристраивание кадров может происходить до тех пор, пока полное число переданных байт не превысит 1518. Пакетная перегруженность уменьшать вероятность образования коллизий, поскольку перегруженный кадр может испытать коллизию только на этапе передачи первого своего оригинального кадра, включая расширение носителя, что, безусловно, увеличивает производительность сети, особенно при больших нагрузках.

15. БЕСПРОВОДНЫЕ ЛОКАЛЬНЫЕ СЕТИ (Wi-Fi)

Беспроводные локальные сети сегодня рассматриваются как дополнение к проводным сетям, а не как конкурентное решение. Отношение к беспроводным локальным сетям не всегда было таковым, в середине 90-х было популярно мнение, в соответствии с которым все большее число локальных сетей будет переходить на беспроводные технологии. Преимущество беспроводных локальных сетей очевидно - их проще и дешевле разворачивать и модифицировать, так как вся громоздкая кабельная и инфраструктура оказывается излишней. Еще одно преимущество — обеспечение мобильности пользователей. Однако за эти преимущества беспроводные сети расплачиваются большим перечнем проблем, которые несет с собой неустойчивая и непредсказуемая беспроводная среда.

Бурное развитие технологии беспроводной связи привело к тому, что пользователи, не успев привыкнуть к одному стандарту, вынуждены переходить на другой, предлагающий еще более высокие скорости передачи. Речь, конечно же, идет о семействе протоколов беспроводной связи, известном как IEEE 802.11, куда входят следующие протоколы: 802.11, 802.11b, 802.11b+, 802.11a, 802.11g. В последнее время стали говорить и о расширении протокола 802.11g.

Различные типы беспроводных сетей отличаются друг от друга и радиусом действия, и поддерживаемыми скоростями соединения, и технологией кодирования данных. Так, стандарт IEEE 802.11b предусматривает максимальную скорость соединения 11 Мбит/с, стандарт IEEE 802.11b+ - 22 Мбит/с, стандарты IEEE 802.11g и 802.11a - 54 Мбит/с.

Будущее стандарта 802.11a довольно туманно. Наверняка в России и в Европе этот стандарт не получит широкого распространения, да и в США, где он сейчас используется, скорее всего, в ближайшее время произойдет переход на альтернативные стандарты. А вот новый стандарт 802.11g имеет значительные шансы завоевать признание во всем мире. Другое преимущество нового стандарта 802.11g заключается в том, что он полностью совместим со стандартами 802.11b и

802.11b+, то есть любое устройство, поддерживающее стандарт 802.11g, будет работать (правда, на меньших скоростях соединения) и в сетях стандарта 802.11b/b+, а устройство, поддерживающее стандарт 802.11b/b+ — в сетях стандарта 802.11g, хотя и с меньшей скоростью соединения.

Совместимость стандартов 802.11g и 802.11b/b+ обусловлена, во-первых, тем, что они предполагают использование одного и того же частотного диапазона, а во-вторых, что все режимы, предусмотренные в протоколах 802.11b/b+, реализованы и в стандарте 802.11g. Поэтому стандарт 802.11b/b+ можно рассматривать как подмножество стандарта 802.11g.

15.1. Стек протоколов IEEE 802.11

Естественно, что стек протоколов стандарта IEEE 802.11 соответствует общей структуре стандартов комитета 802, то есть состоит из физического уровня и уровня MAC, над которыми работает уровень LLC. Как и у всех технологий семейства 802, технология 802.11 определяется нижними двумя уровнями, то есть физическим уровнем и уровнем MAC, а уровень LLC выполняет свои стандартные общие для всех технологий LAN функции. Так как искажения кадров в беспроводной среде более вероятны, чем в проводной, уровень LLC должен, скорее всего, использоваться в режиме LLC2. Но это уже не зависит от технологии 802.11, режим работы уровня LLC выбирается протоколами верхних уровней.

Структура стека протоколов IEEE 802.11 показана на рис. 15.1.

На физическом уровне существует несколько вариантов спецификаций, которые отличаются используемым частотным диапазоном, методом кодирования и как следствие – скоростью передачи данных. Все варианты физического уровня работают с одним и тем же алгоритмом уровня MAC, но некоторые временные параметры уровня MAC зависят от используемого физического уровня.

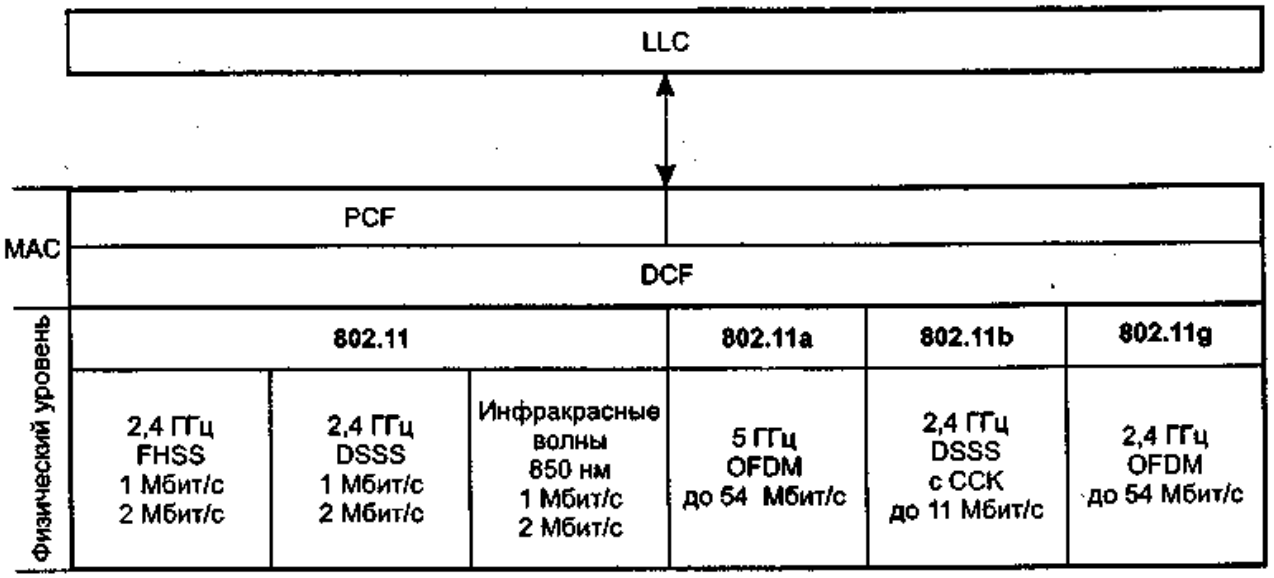


Рисунок 15.1. Стек протоколов IEEE 802.11

Технология уширения спектра

В основе всех беспроводных протоколов семейства 802.11 лежит технология уширения спектра (Spread Spectrum, SS). Данная технология подразумевает, что первоначально узкополосный (в смысле ширины спектра) полезный информационный сигнал при передаче преобразуется таким образом, что его спектр оказывается значительно шире спектра первоначального сигнала. То есть спектр сигнала как бы «размазывается» по частотному диапазону. Одновременно с уширением спектра сигнала происходит и перераспределение спектральной энергетической плотности сигнала — энергия сигнала также «размазывается» по спектру. В результате максимальная мощность преобразованного сигнала оказывается значительно ниже мощности исходного сигнала. При этом уровень полезного информационного сигнала может в буквальном смысле сравниваться с уровнем естественного шума. В результате сигнал становится в каком то смысле «невидимым» — он просто теряется на уровне естественного шума.

Собственно, именно в изменении спектральной энергетической плотности сигнала и заключается идея уширения спектра. Дело в том, что если подходить к проблеме передачи данных традиционным способом, то есть так, как это делается

в радиоэфире, где каждой радиостанции отводится свой диапазон вещания, то мы неизбежно столкнемся с проблемой, что в ограниченном радиодиапазоне, предназначенном для совместного использования, невозможно «уместить» всех желающих. Поэтому необходимо найти такой способ передачи информации, при котором пользователи могли бы сосуществовать в одном частотном диапазоне и при этом не мешать друг другу. Именно эту задачу и решает технология уширения спектра.

В 1997 году комитетом **802.11** был принят стандарт, который определял функции уровня MAC вместе с ***тремя вариантами физического уровня***, которые обеспечивают передачу данных со скоростями 1 и 2 Мбит/с.

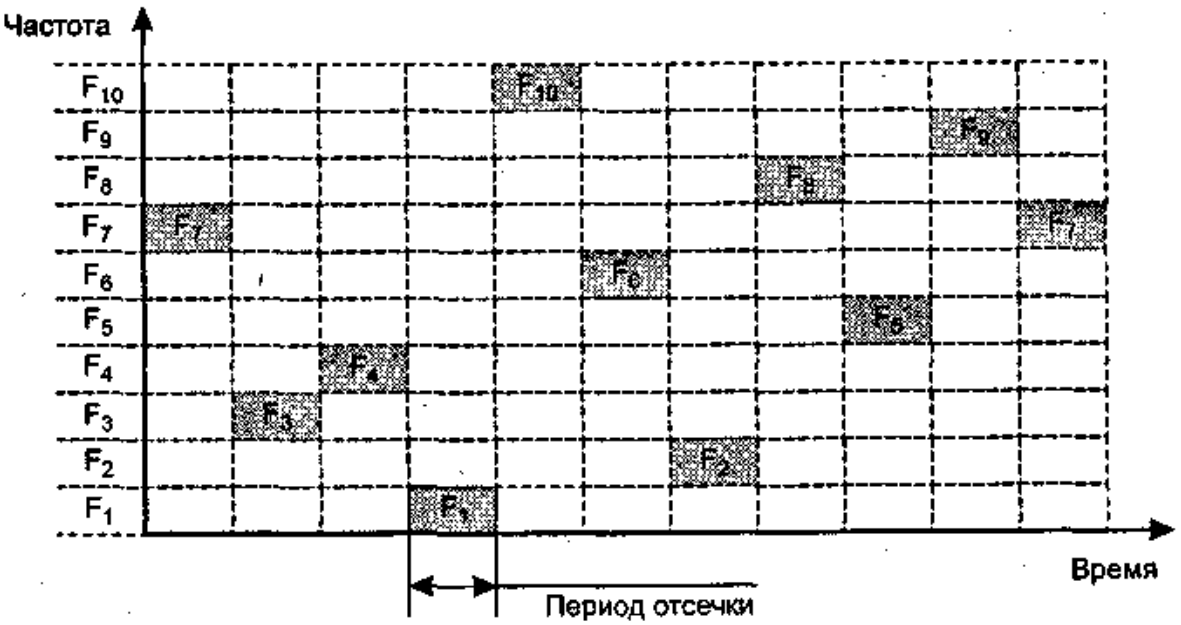
В первом варианте средой являются ***инфракрасные волны*** диапазона 850 нм, которые генерируются либо полупроводниковым лазерным диодом, либо светодиодом (LED). Так как инфракрасные волны не проникают через стены, область покрытия LAN ограничивается зоной прямой видимости.

Во втором варианте в качестве передающей среды используется ***микроволновый диапазон*** 2,4 ГГц, который в соответствии с рекомендациями ITU в большинстве стран не лицензируется. Этот вариант основан на методе FHSS. В методе FHSS каждый узкий канал имеет ширину 1 МГц. Частотная манипуляция (FSK) с двумя состояниями сигнала (частотами) дает скорость 1 Мбит/с, с четырьмя состояниями — 2 Мбит/с. Количество каналов и частота переключения между каналами настраиваются, так что при развертывании беспроводной локальной сети можно учитывать особенности регулирования спектра частот конкретной страны.

Идею метода **расширения спектра скачкообразной перестройкой частоты** (Frequency Hopping Spread Spectrum, FHSS) иллюстрирует рис. 15.2.

В течение определенного фиксированного интервала времени передача ведется на неизменной несущей частоте. На каждой несущей частоте для передачи дискретной информации применяются стандартные методы модуляции, такие как FSK или PSK. Для того чтобы приемник синхронизировался с передатчиком, для обозначения начала каждого периода передачи в течение

некоторого времени передаются синхробиты. Так что полезная скорость этого метода кодирования оказывается меньше из-за постоянных накладных расходов на синхронизацию. Несущая частота меняется в соответствии с номерами частотных подканалов, вырабатываемых алгоритмом псевдослучайных чисел. Псевдослучайная последовательность зависит от некоторого параметра, который называют **начальным числом.** Если приемнику и передатчику известны алгоритм и значение начального числа, то они меняют частоты в одинаковой последовательности, называемой **последовательностью псевдослучайной перестройки частоты.**



Последовательность перестройки частот: F_7-F_3-F_4-F_1-F_{10}-F_6-F_2-F_8-F_5-F_9

Рисунок 15.2. Расширение спектра скачкообразной перестройкой частоты

Если частота смены подканалов ниже, чем скорость передачи данных в канале, то такой режим называют медленным **расширением спектра** (рис. 15.3, *а*)*;* в противном случае мы имеем дело с **быстрым расширением спектра** (рис. 15.3, *б*).

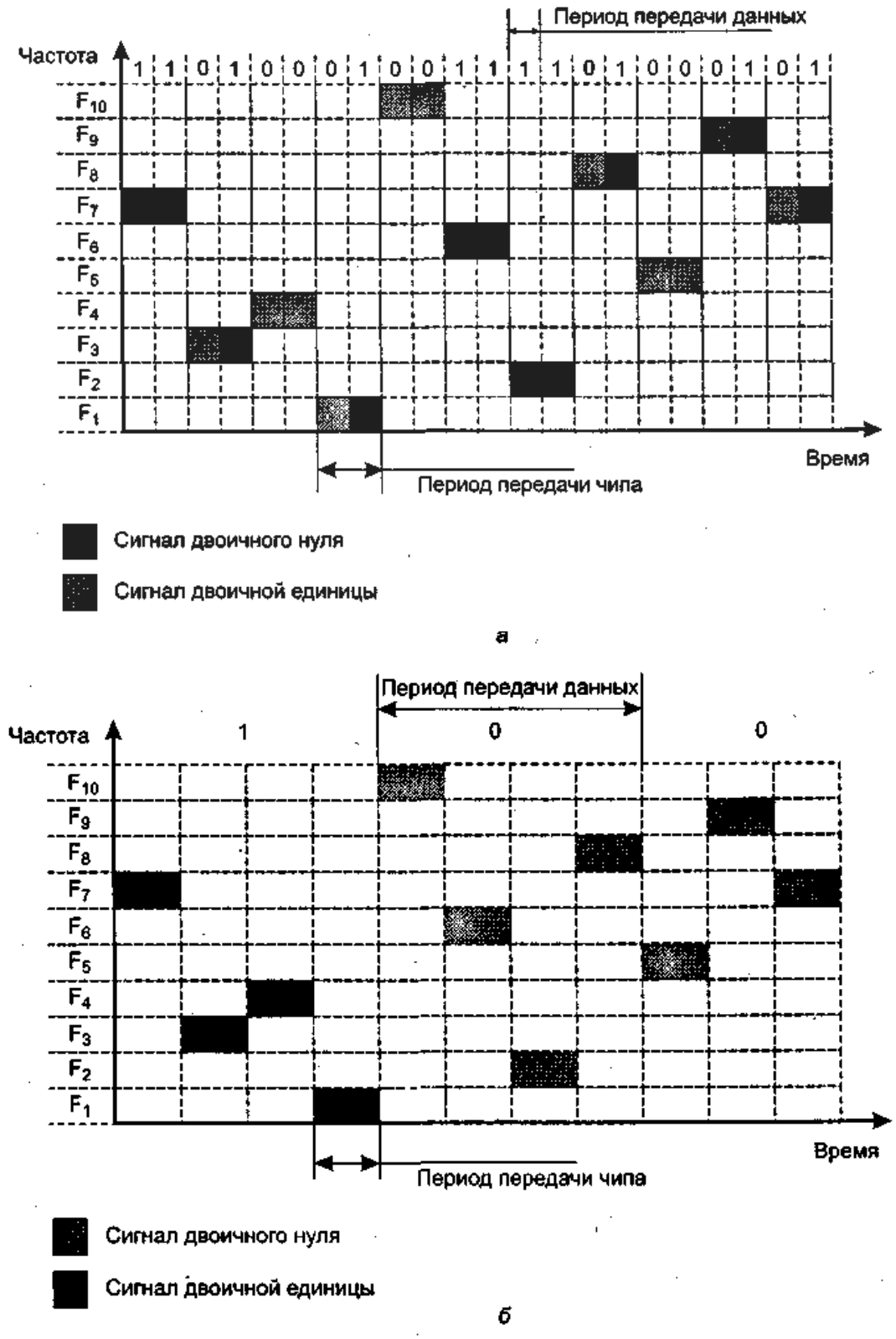


Рисунок 15.3 Соотношение между скоростью передачи данных и частотой смены подканалов:

а — скорость передачи данных выше чиповой скорости,

б — скорость передачи данных ниже чиповой скорости.

Третий вариант, в котором используется тот же ***микроволновый диапазон,*** основан на методе DSSS, где в качестве последовательности чипов применяется 11-битный код 10110111000. Каждый бит кодируется путем двоичной фазовой (1 Мбит/с) или квадратурной фазовой (2 Мбит/с) манипуляции.

В методе **прямого последовательного расширения спектра** (Direct Sequence Spread Spectrum, DSSS) также используется весь частотный диапазон, выделенный для одной беспроводной линии связи. В отличие от метода FHSS весь частотный диапазон занимается не за счет постоянных переключений с частоты на частоту, а за счет того, что каждый бит информации заменяется N битами, так что тактовая скорость передачи сигналов увеличивается в N раз. А это, в свою очередь, означает, что спектр сигнала также расширяется в N раз. Достаточно соответствующим образом выбрать скорость передачи данных и значение N, чтобы спектр сигнала заполнил весь диапазон.

Цель кодирования методом DSSS та же, что методом FHSS — повышение устойчивости к помехам. Узкополосная помеха будет искажать только определенные частоты спектра сигнала, так что приемник с большой степенью вероятности сможет правильно распознать передаваемую информацию.

Код, которым заменяется двоичная единица исходной информации, называется **расширяющей последовательностью**, а каждый бит такой последовательности — **чипом.** Соответственно, скорость передачи результирующего кода называют **чиповой скоростью.** Двоичный нуль, кодируется инверсным значением расширяющей последовательности. Приемники должны знать расширяющую последовательность, которую использует передатчик, чтобы понять передаваемую информацию.

Примером значения расширяющей последовательности является *последовательность Баркера* (Barker), которая состоит из 11 бит: 10110111000 (рис.15.4). Если передатчик использует эту последовательность, то передача трех битов 110 ведет к передаче следующих битов:

10110111000 10110111000 01001000111.

Последовательность Баркера позволяет приемнику быстро синхронизироваться с передатчиком, то есть надежно выявлять начало последовательности. Приемник определяет такое событие, поочередно сравнивая получаемые биты с образцом последовательности. Действительно, если сравнить последовательность Баркера с такой же последовательностью, но сдвинутой на один бит влево или вправо, то мы получим меньше половины совпадений значений битов. Значит, даже при искажении нескольких битов с большой долей вероятности приемник правильно определит начало последовательности, а значит, сможет правильно интерпретировать получаемую информацию.

Метод DSSS в меньшей степени защищен от помех, чем метод быстрого расширения спектра, так как, мощная узкополосная помеха влияет на часть спектра, а значит, и на результат распознавания единиц или нулей.

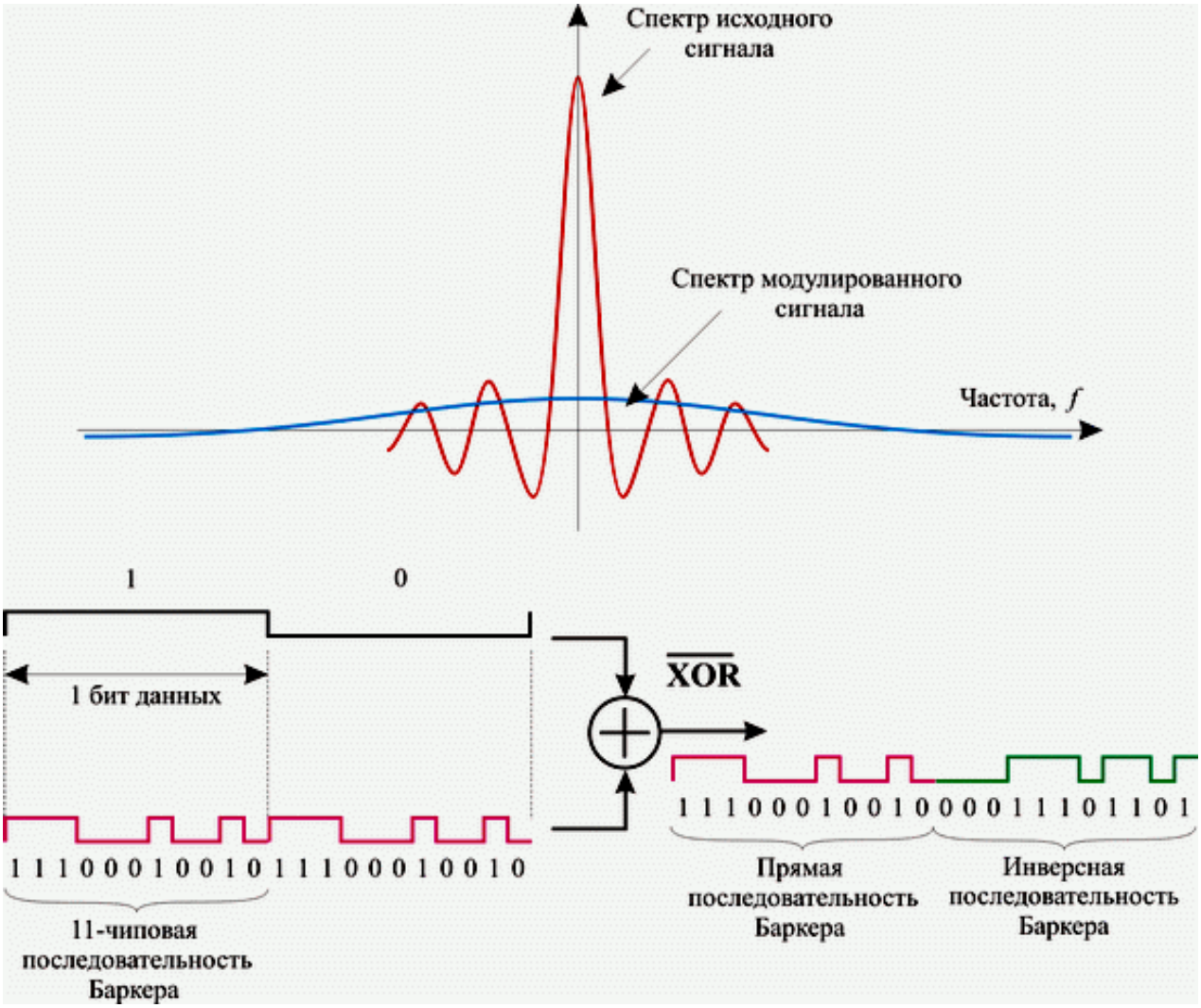


Рисунок 15.4. Метод прямого последовательного расширения спектра .

Скорость 1 Мбит/с

Для модуляции синусоидального несущего сигнала (процесс, необходимый для информационного наполнения несущего сигнала) используется относительная двоичная фазовая модуляция (Differential Binary Phase Shift Key, DBPSK). При этом кодирование информации происходит за счет сдвига фазы синусоидального сигнала по отношению к предыдущему состоянию сигнала. Двоичная фазовая модуляция предусматривает два возможных значения сдвига фазы — 0 и π. Тогда логический нуль может передаваться синфазным сигналом (сдвиг по фазе равен 0), а единица — сигналом, который сдвинут по фазе на π.

Скорость 2 Мбит/с

Информационная скорость 1 Мбит/с является обязательной в стандарте IEEE 802.11 (Basic Access Rate), но опционально возможна и скорость в 2 Мбит/с (Enhanced Access Rate). Для передачи данных на такой скорости используется та же технология DSSS с 11-чиповыми кодами Баркера, но для модуляции несущего колебания применяется относительная квадратурная фазовая модуляция (Differential Quadrature Phase Shiftey). При относительной квадратурной фазовой модуляции сдвиг фаз может принимать четыре различных значения: 0, π/2, π и 3π/2. Используя четыре различных состояния сигнала, можно в одном дискретном состоянии закодировать последовательность двух информационных бит (дибит) и тем самым в два раза повысить информационную скорость передачи. К примеру, дибиту 00 может соответствовать сдвиг фазы, равный 0; дибиту 01 — сдвиг фазы, равный π/2; дибиту 11 — сдвиг фазы, равный π; дибиту 10 — сдвиг фазы, равный 3π/2.

В 1999 году были приняты еще два варианта физического уровня: **802.11a** и **802.11b.**

Спецификация 802.11a обеспечивает повышение скорости за счет более высокого диапазона частот (5 ГГц). Для этого задействуются 300 МГц из этого диапазона, ортогональное частотное мультиплексирование (OFDM) и прямая коррекция ошибок (FEC). Скорости передачи данных составляют 6, 9, 12, 18, 24, 36, 48 и 54 Мбит/с. Диапазон 5 ГГц спецификации 802.11a пока мало «населен» и

обеспечивает высокие скорости передачи данных. Однако его использование связано с двумя проблемами. Во-первых, оборудование для этих частот пока еще слишком дорогое, во-вторых, в некоторых странах частоты этого диапазона подлежат лицензированию.

В спецификации 802.11b института IEEE по-прежнему используется диапазон 2,4 ГГц, что позволяет задействовать более дешевое оборудование. Для повышения скорости до 11 Мбит/с, которая сопоставима со скоростью классического стандарта Ethernet, здесь применяется более эффективный метод DSSS, использующий технику Complementary Code Keying (CCK).

Протокол IEEE 802.11b, принятый в июле 1999 года, является своего рода расширением базового протокола 802.11 и кроме скоростей 1 и 2 Мбит/с предусматривает скорости 5,5 и 11 Мбит/с. Для работы на скоростях 1 и 2 Мбит/с используются технология уширения спектра с использованием кодов Баркера, а для скоростей 5,5 и 11 Мбит/с используются так называемые комплементарные коды (Complementary Code Keying, CCK).

CCK-последовательности

Комплементарные коды или CCK-последовательности обладают тем свойством, что сумма их автокорреляционных функций для любого циклического сдвига, отличного от нуля, всегда равна нулю. Если система связи, использующая комплементарное кодирование, работает в условиях многопутевого распространения сигналов, то в идеале межсимвольная интерференция (вызванная наложением сигналов с задержками распространения) должна отсутствовать, поскольку сумма их автокорреляционных функций равна нулю.

В стандарте IEEE 802.11b речь идет о комплексных комплементарных 8-чиповых последовательностях, определенных на множестве комплексных элементов.

Используя множество комплексных элементов {1, –1, j, –j} можно сформировать восемь одинаковых по модулю, но отличающихся по фазе комплексных чисел. То есть, элементы 8-чиповой CCK-последовательности могут принимать одно из следующих восьми значений: 1, –1, j, –j, 1+j, 1–j, –1+j, –1–j.

Основное отличие CCK-последовательностей от рассмотренных ранее кодов Баркера заключается в том, что существует не строго заданная последовательность, посредством которой можно было кодировать либо логический нуль, либо единицу, а целый набор последовательностей. Учитывая, что каждый элемент 8-чиповой последовательности может принимать одно из восьми значений в зависимости от значения фазы, ясно, что можно скомбинировать 8^8=16777216 вариантов последовательностей, однако, не все они будут комплементарными. Но даже с учетом требования комплементарности можно сформировать достаточно большое число разных CCK-последовательностей. Это обстоятельство позволяет кодировать в одном передаваемом символе несколько информационных бит и тем самым повысить информационную скорость передачи.

Вообще говоря, использование CCK-кодов позволяет кодировать 8 бит на один символ при скорости 11 Мбит/с и 4 бит на символ при скорости 5,5 Мбит/с. При этом в обоих случаях символьная скорость передачи составляет $1{,}385\times10^6$ символов в секунду (11/8 = 5,5/4 = 1,385), а учитывая, что каждый символ задается 8-чиповой последовательностью, получаем, что в обоих случаях скорость следования отдельных чипов составляет 11×10^6 чипов в секунду. Соответственно, и ширина спектра сигнала как при скорости 11 Мбит/с и 5,5 Мбит/с составляет 22 МГц.

Двоичное пакетное сверточное кодирование PBCC

Для дальнейшего рассмотрения протокола 802.11b/b+ нам предстоит ознакомиться с еще одним типом кодирования — так называемым двоичным пакетным сверточным кодированием (Packet Binary Convolutional Coding, PBCC).

Идея сверточного кодирования заключается в следующем. Входящая последовательность информационных бит преобразуется в специальном сверточном кодере таким образом, чтобы каждому входному биту соответствовало более одного выходного. То есть сверточный кодер добавляет определенную избыточную информацию к исходной последовательности. Если, к примеру, каждому входному биту соответствует два выходных, то говорят о

сверточном кодировании со скоростью r = 1/2. Если же каждым двум входным битам соответствует три выходных, то скорость сверточного кодирования будет составлять уже 2/3.

Любой сверточный кодер строится на основе нескольких последовательно связанных запоминающих ячеек и логических элементов, связывающих эти ячейки между собой. Количество запоминающих ячеек определяет количество возможных состояний кодера. Если, к примеру, в сверточном кодере используется шесть запоминающих ячеек, то в кодере хранится информация о шести предыдущих состояниях сигнала, а с учетом значения входящего бита получим, что в таком кодере используется семь бит входной последовательности. Такой сверточный кодер называется кодером на семь состояний (K = 7).

Выходные биты, формируемые в сверточном кодере, определяются значениями входного бита и битами, хранимыми в запоминающих ячейках, то есть значение каждого формируемого выходного бита зависит не только от входящего информационного бита, но и от нескольких предыдущих битов.

В технологии PBCC используются сверточные кодеры на семь состояний (K = 7) со скоростью r=1/2. Главным достоинством сверточных кодеров является помехоустойчивость формируемой ими последовательности. Дело в том, что при избыточности кодирования даже в случае возникновения ошибок приема исходная последовательность бит может быть безошибочно восстановлена. Для восстановления исходной последовательности битов на стороне приемника применяется декодер Витерби.

При скорости передачи 5,5 Мбит/с для модуляции дибита, формируемого сверточным кодером, используется двоичная фазовая модуляция, а при скорости 11 Мбит/с — квадратурная фазовая модуляция. При этом для скорости 11 Мбит/с в каждом символе кодируется по одному входному биту и скорость передачи бит соответствует скорости передачи символов, а при скорости 5,5 Мбит/с скорость передачи битов равна половине скорости передачи символов (поскольку каждому входному биту в данном случае соответствует два выходных символа). Поэтому и

для скорости 5,5 Мбит/с, и для скорости 11 Мбит/с символьная скорость составляет 11×10^6 символов в секунду.

Для скорости 22 Мбит/с по сравнению с уже рассмотренной нами схемой PBCC передача данных имеет две особенности. Прежде всего, используется фазовая 8-позиционная фазовая модуляция (8-PSK), то есть фаза сигнала может принимать восемь различных значений, что позволяет в одном символе кодировать уже 3 бита. Кроме того, в схему кроме сверточного кодера добавлен пунктурный кодер (Puncture). Смысл такого решения довольно прост: избыточность сверточного кодера, равная 2 (на каждый входной бит приходится два выходных), достаточно высока и при определенных условиях помеховой обстановки является излишней, поэтому можно уменьшить избыточность, чтобы, к примеру, каждым двум входным битам соответствовало три выходных.

Разобравшись с принципом работы пунктурного кодера, вернемся к рассмотрению кодирования PBCC на скорости 22 Мбит/с в протоколе 802.11b+.

В сверточный кодер (K = 7, R = 1/2) данные поступают со скоростью 22 Мбит/с. После добавления избыточности в сверточном кодере биты со скоростью потока 44 Мбит/с поступают в пунктурный кодер 4:3, в котором избыточность уменьшается так, чтобы на каждые четыре входных бита приходилось три выходных. Следовательно, после пунктурного кодера скорость потока составит уже 33 Мбит/с (не информационная, а общая скорость с учетом добавленных избыточных битов). Полученная в результате последовательность направляется в фазовый модулятор 8-PSK, где каждые три бита упаковываются в один символ. При этом скорость передачи составит 11×10^6 символов в секунду, а информационная скорость — 22 Мбит/с (рис. 15.5).

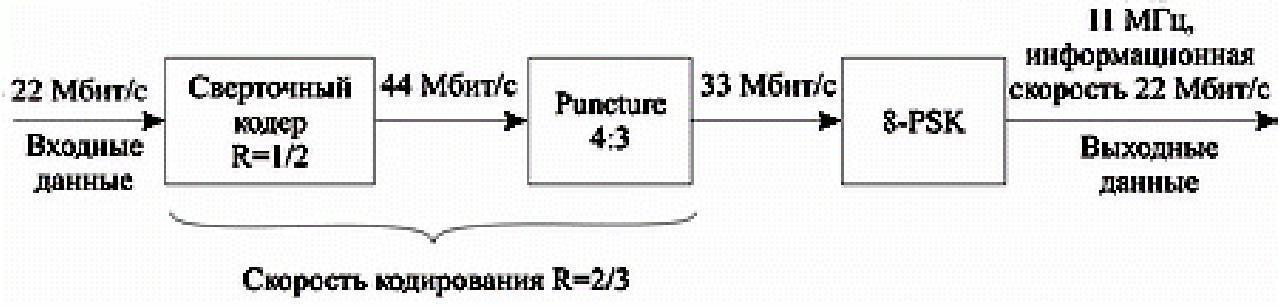


Рисунок 15.5. Реализация скорости 22 Мбит/с в протоколе 802.11g.

На физическом уровне к MAC-кадрам (MPDU) добавляется заголовок физического уровня, состоящий из преамбулы и собственно PLCP-заголовка (рис.). Преамбула содержит стартовую синхро-последовательность (SYNC) для настройки приемника и 16-битный код начала кадра (SFD) — число F3A0. PLCP-заголовок включает поля SIGNAL (информация о скорости и типе модуляции), SERVICE (дополнительная информация, в том числе о применении высокоскоростных расширений и PBSS-модуляции) и LENGTH (время в микросекундах, необходимое для передачи следующей за заголовком части кадра). Все три поля заголовка защищены 16-битной контрольной суммой CRC.

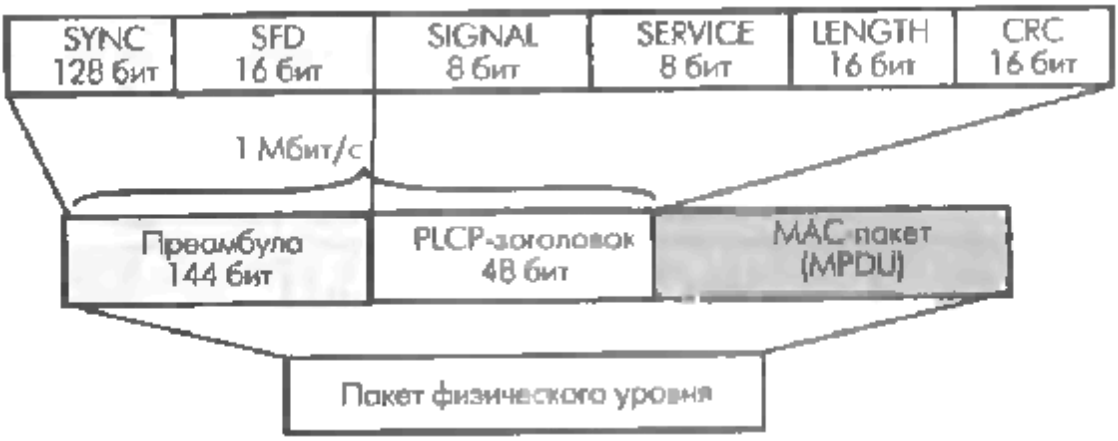


Рисунок 15.6. Структура кадров физического уровня сети 802.11

В стандарте IEEE 802.11b предусмотрено два типа заголовков: длинный и короткий. Они отличаются длиной синхро-последовательности (128 и 56 бит), способом ее генерации, а также тем, что символ начала кадра в коротком заголовке передается в обратном порядке. Кроме того, если все поля длинного заголовка передаются со скоростью 1 Мбит/с, то при коротком заголовке преамбула транслируется на скорости 1 Мбит/с, другие поля заголовка — со скоростью 2 Мбит/с. Остальную часть кадра можно передавать на любой из допустимых стандартом скоростей передачи, указанных в полях SIGNAL и SERVICE. Короткие заголовки физического уровня предусмотрены спецификацией IEEE 802.1b для увеличения пропускной способности сети.

Еще один стандарт для физического уровня разработан группой **802.llg** института IEEE летом 2003 года. В нем также задействован диапазон 2,4 ГГц, но со скоростью передачи данных до 54 Мбит/с. В этой спецификации используется ортогональное частотное мультиплексирование (OFDM). До недавнего времени в США в диапазоне 2,4 ГГц разрешалось работать только за счет расширения спектра. Снятие этого ограничения дало импульс новым разработкам, в результате появилась новая высокоскоростная беспроводная технология. Для обратной совместимости с 802.11b поддерживается также техника ССК.

Стандарт IEEE 802.11g является логическим развитием стандарта 802.11b/b+ и предполагает передачу данных в том же частотном диапазоне, но с более высокими скоростями. Кроме того, стандарт 802.11g полностью совместим с 802.11b, то есть любое устройство 802.11g должно поддерживать работу с устройствами 802.11b. Максимальная скорость передачи в стандарте 802.11g составляет 54 Мбит/с.

При разработке стандарта 802.11g рассматривались несколько конкурирующих технологий: метод ортогонального частотного разделения OFDM, предложенный к рассмотрению компанией Intersil, и метод двоичного пакетного сверточного кодирования PBCC, опционально реализованный в стандарте 802.11b и предложенный компанией Texas Instruments. В результате стандарт 802.11g основан на компромиссном решении: в качестве базовых применяются технологии OFDM и CCK, а опционально предусмотрено использование технологии PBCC.

Ортогональное частотное разделение каналов с мультиплексированием

В стандарте 802.11b с максимальной скоростью передачи 11 Мбит/с при использовании CCK-кодов схемы компенсации межсимвольной интерференции вполне успешно справляются с возложенной на них задачей, но при более высоких скоростях такой подход становится неприемлемым.

Поэтому при более высоких скоростях передачи применяется принципиально иной метод кодирования данных – ортогональное частотное разделение каналов с мультиплексированием (Orthogonal Frequency Division

Multiplexing, OFDM). Идея данного метода заключается в том, что поток передаваемых данных распределяется по множеству частотных подканалов и передача ведется параллельно на всех этих подканалах. При этом высокая скорость передачи достигается именно за счет одновременной передачи данных по всем каналам, а скорость передачи в отдельном подканале может быть и невысокой.

Поскольку в каждом из частотных подканалов скорость передачи данных можно сделать не слишком высокой, это создает предпосылки для эффективного подавления межсимвольной интерференции.

При частотном разделении каналов необходимо, чтобы ширина отдельного канала была, с одной стороны, достаточно узкой для минимизации искажения сигнала в пределах отдельного канала, а с другой — достаточно широкой для обеспечения требуемой скорости передачи. Кроме того, для экономного использования всей полосы канала, разделяемого на подканалы, желательно как можно более плотно расположить частотные подканалы, но при этом избежать межканальной интерференции, чтобы обеспечить полную независимость каналов друг от друга. Частотные каналы, удовлетворяющие перечисленным требованиям, называются ортогональными. Несущие сигналы всех частотных подканалов (а точнее, функции, описывающие эти сигналы) ортогональны друг другу.

Диаметр сети 802.11 зависит от многих параметров, в том числе и от диапазона частот. Обычно диаметр беспроводной локальной сети находится в пределах от 100 до 300 м.

Уровень MAC выполняет в беспроводных сетях больше функций, чем в проводных сетях. Функции уровня MAC в стандарте 802.11 включают:

- доступ к разделяемой среде;
- обеспечение мобильности станций при наличии нескольких базовых станций;
- обеспечение безопасности, эквивалентной безопасности проводных локальных сетей.

15.2.Топологии локальных сетей стандарта 802.11

Станции могут использовать разделяемую среду для того, чтобы передавать данные:

- непосредственно друг другу в пределах одной BSS-сети;
- в пределах одной BSS-сети транзитом через точку доступа;
- между разными BSS-сетями через две точки доступа и распределенную систему;
- между BSS-сетью и проводной локальной сетью через точку доступа, распределенную систему и портал (Функции портала стандартом не детализируются, это может быть коммутатор или маршрутизатор).

Стандарт 802.11 поддерживает два типа топологий локальных сетей: с базовым и с расширенным наборами услуг.

Сеть с базовым набором услуг (Basic Service Set, BSS) режим Ad Hoc (рис.15.7), который называют также режимом Peer to Peer (точка-точка), станции непосредственно взаимодействуют друг с другом. Для этого режима нужен минимум оборудования: каждая станция должна быть оснащена беспроводным адаптером. При такой конфигурации не требуется создания сетевой инфраструктуры. Основными недостатками режима Ad Hoc являются ограниченный диапазон действия возможной сети и невозможность подключения к внешней сети (например, к Интернету). Базовая станция отсутствует, узлы взаимодействуют друг с другом непосредственно (рис.15.7). Для того чтобы войти в BSS-сеть, станция должна выполнить процедуру присоединения.

BSS-сети не являются традиционными сотами в отношении зон покрытия, они могут находиться друг от друга на значительном расстоянии, а могут частично или полностью перекрываться - стандарт 802.11 оставляет здесь свободу для проектировщика сети.

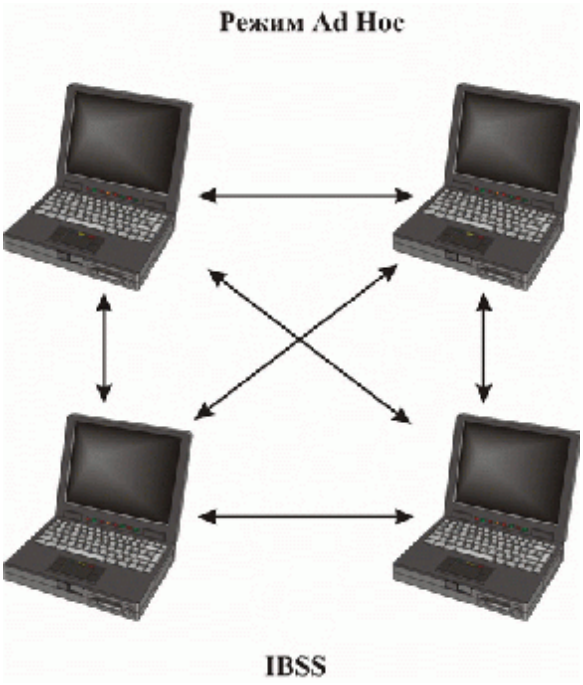


Рисунок 15.7. Режим функционирования Ad Hoc

В режиме **Infrastructure Mode** (рис.15.8) станции взаимодействуют друг с другом не напрямую, а через точку доступа (Access Point), которая выполняет в беспроводной сети роль своеобразного концентратора (аналогично тому, как это происходит в традиционных кабельных сетях). Рассматривают два режима взаимодействия с точками доступа — BSS (Basic Service Set) и ESS (Extended Service Set). В режиме BSS все станции связываются между собой только через точку доступа, которая может выполнять также роль моста к внешней сети. В расширенном режиме ESS существует инфраструктура нескольких сетей BSS, причем сами точки доступа взаимодействуют друг с другом, что позволяет передавать трафик от одной BSS к другой. Между собой точки доступа соединяются с помощью либо сегментов кабельной сети, либо радиомостов.

В сетях, обладающих инфраструктурой, некоторые станции сети являются базовыми, или, в терминологии 802.11, **точками доступа (Access Point, AP)**. Станция, которая выполняет функции AP, является членом какой-нибудь BSS-сети (рис.15.8). Все базовые станции сети связаны между собой с помощью **распределенной системы (Distribution System, DS)**, в качестве которой может

использоваться - та же среда (то есть радио- или инфракрасные волны), что и для взаимодействия между станциями, или же отличная от нее, например проводная.

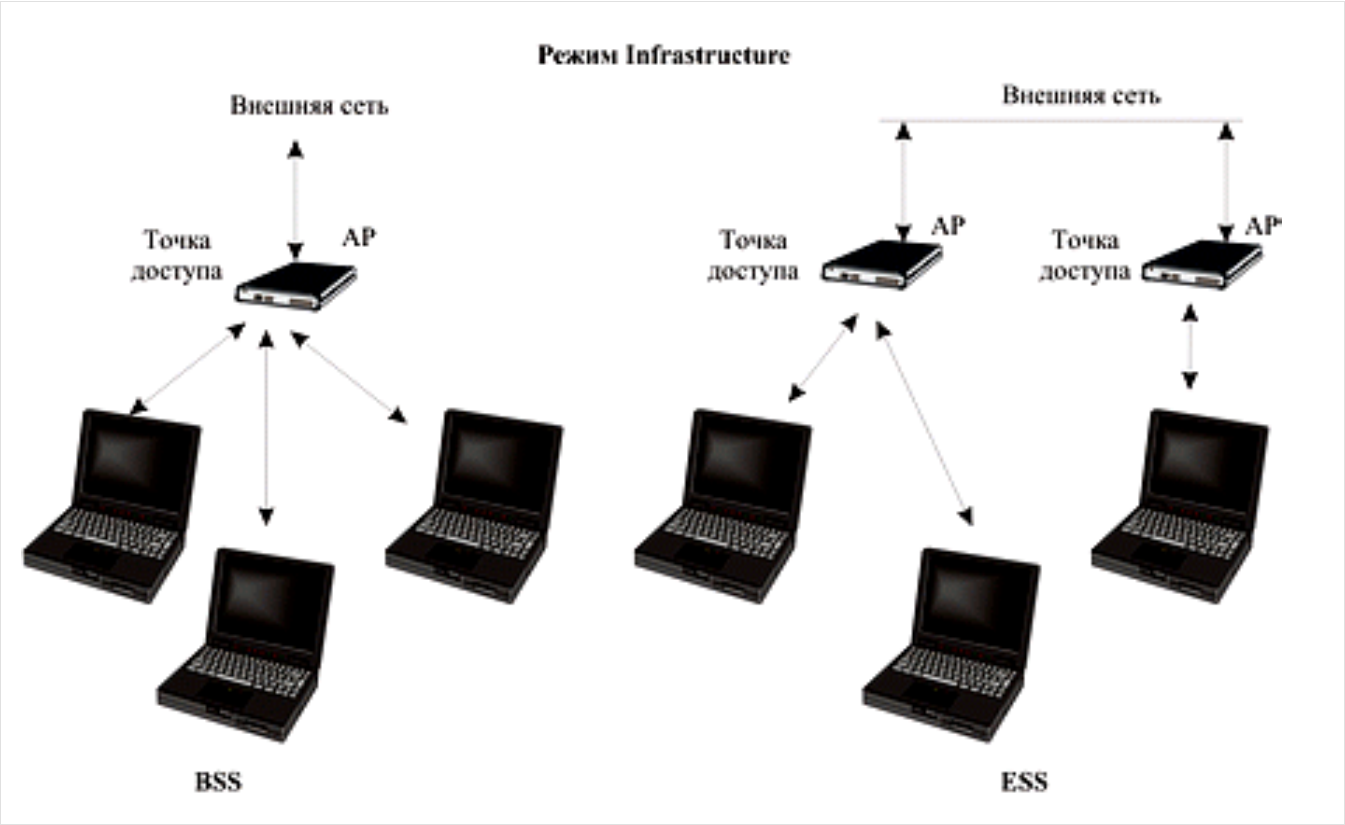


Рисунок 15.8. Режим функционирования Infrastructure Mode.

Точки доступа вместе с распределенной системой поддерживают **службу распределенной системы (Distribution System Service, DSS)**. Задачей DSS является передача пакетов между станциями, которые по каким-то причинам не могут или не хотят взаимодействовать между собой непосредственно. Наиболее очевидной причиной использования DSS является принадлежность станций разным BSS-сетям. В этом случае они передают кадр своей точке доступа, которая через DS передает его точке доступа, обслуживающей BSS-сеть со станцией назначения.

Сеть с расширенным набором услуг **(Extended Service Set, ESS**) состоит из нескольких BSS-сетей, объединенных распределенной средой.

ESS-сеть обеспечивает станциям мобильность — они могут переходить из одной BSS-сети в другую. Эти перемещения обеспечиваются функциями уровня MAC рабочих и базовых станций, поэтому они совершенно прозрачны для уровня

LLC. ESS-сеть может также взаимодействовать с проводной локальной сетью. Для этого в распределенной системе должен присутствовать портал.

Распределенный режим доступа DCF

На первый взгляд организовать совместный доступ к среде передачи данных достаточно просто. Для этого необходимо лишь обеспечить, чтобы все узлы передавали данные только тогда, когда среда является свободной, то есть когда ни один из узлов не производит передачу данных. Однако такой механизм неизбежно приведет к коллизиям, поскольку велика вероятность того, что два или более узлов одновременно, пытаясь получить доступ к среде передачи данных, решат, что среда свободна и начнут одновременную передачу. Именно поэтому необходимо разработать алгоритм, способный снизить вероятность возникновения коллизий и в то же время гарантировать всем узлам сети равноправный доступ к среде передачи данных.

Одним из вариантов организации такого равноправного доступа к среде передачи данных является **функция распределенной координации (DCF)**. Эта функция основана на **методе коллективного доступа с обнаружением несущей и механизмом избежания коллизий (Carrier Sense Multiple Access/Collision Avoidance, CSMA/CA)**. При такой организации каждый узел, прежде чем начать передачу, «прослушивает» среду, пытаясь обнаружить несущий сигнал, и только при условии, что среда свободна, может начать передачу данных.

Однако, как мы уже отмечали, в этом случае велика вероятность возникновения коллизий: когда два или более узлов сети одновременно (или почти одновременно) решат, что среда свободна, и начнут предавать данные. Для того чтобы снизить вероятность возникновения подобных ситуаций, используется механизм **избежания коллизий (Collision Avoidance, CA)**. Суть данного механизма заключается в следующем. Каждый узел сети, убедившись, что среда свободна, прежде чем начать передачу, выжидает в течение определенного промежутка времени. Этот промежуток является случайным и складывается из двух составляющих: **обязательного промежутка DIFS (DCF Interframe Space)** и выбираемого случайным образом **промежутка обратного отсчета (backoff time)**.

В результате каждый узел сети перед началом передачи выжидает в течение случайного промежутка времени, что, естественно, значительно снижает вероятность возникновения коллизий, поскольку вероятность того, что два узла сети будут выжидать в течение одного и того же промежутка времени, чрезвычайно мала.

Для того чтобы гарантировать всем узлам сети равноправный доступ к среде передачи данных, необходимо соответствующим образом определить алгоритм выбора длительности промежутка обратного отсчета (backoff time). Промежуток обратного отсчета хотя и является случайным, но в то же время определяется на основании множества некоторых дискретных промежутков времени, то есть, равен целому числу элементарных временных промежутков, называемых **тайм-слотами (SlotTime)**. Для выбора промежутка обратного отсчета каждый узел сети формирует так называемое **окно конкурентного доступа (Contention Window, CW)**, использующееся для определения количества тайм-слотов, в течение которых станция выжидала перед передачей. Фактически окно CW – это диапазон для выбора количества тайм-слотов, причем минимальной размер окна определяется в 31 тайм-слот, а максимальный размер — в 1023 тайм-слота. Промежуток обратного отсчета определяется как количество тайм-слотов, определяемое исходя из размера окна CW:

$$Backoff\ \ time = Random[CW_{\min}, CW_{\max}] \times SlotTime$$

Размер слота зависит от способа кодирования сигнала; так, для метода FHSS размер слота равен 28 мкс, а для метода DSSS — 1 мкс. Размер слота выбирается таким образом, чтобы он превосходил время распространения сигнала между любыми двумя станциями сети плюс время, затрачиваемое станцией на распознавание занятости среды. Если такое условие соблюдается, то каждая станция сети сумеет правильно распознать начало передачи кадра при прослушивании слотов, предшествующих выбранному ею для передачи слоту.

Когда узел сети пытается получить доступ к среде передачи данных, то после обязательного промежутка ожидания DIFS запускается процедура обратного отсчета, то есть включается обратный отсчет счетчика тайм-слотов

начиная от выбранного значения окна CW. Если в течение всего промежутка ожидания среда оставалась свободной (счетчик обратного отсчета равен нулю), то узел начинает передачу.

После успешной передачи окно CW формируется вновь. Если же за время ожидания передачу начал другой узел сети, то значение счетчика обратного отсчет останавливается и передача данных откладывается. После того как среда станет свободной, данный узел снова начинает процедуру обратного отсчета, но уже с меньшим размером окна CW, определяемого предыдущим значением счетчика обратного отсчета и соответственно с меньшим значением времени ожидания. При этом, очевидно, что чем большее число раз узел откладывает передачу по причине занятости среды, тем выше вероятность того, что в следующий раз он получит доступ к среде передачи данных (рис.15.9).

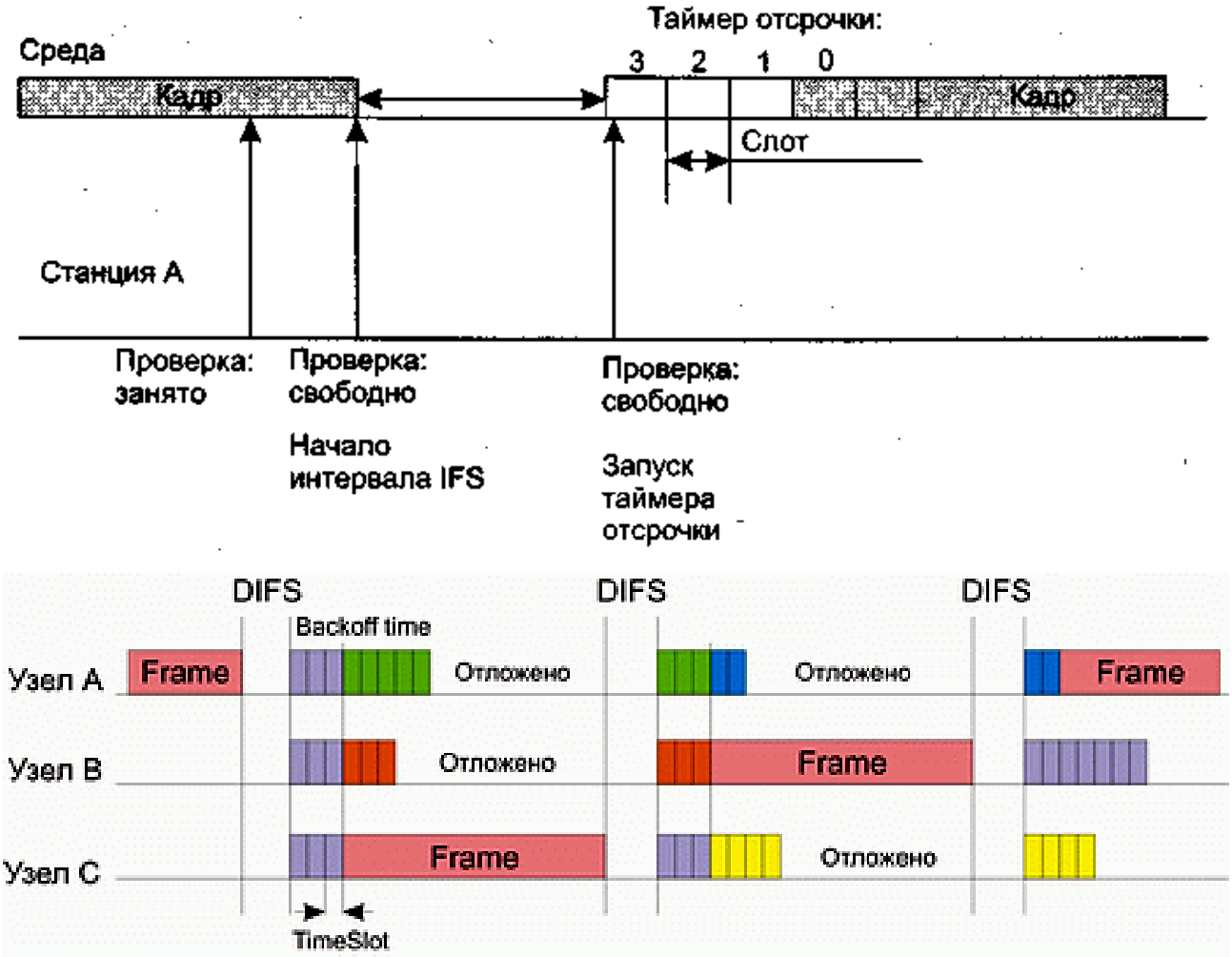


Рисунок 15.9. Реализация доступа к среде передачи данных в методе DCF.

Рассмотренный алгоритм реализации коллективного доступа к среде передачи данных гарантирует равноправный доступ всех узлов сети к среде. Однако при таком подходе вероятность возникновения коллизий хотя и мала, но все-таки существует. Понятно, что снизить вероятность возникновения коллизий можно путем увеличения максимального размера формируемого окна CW. В то же время это увеличит времена задержек при передаче и тем самым снизит производительность сети. Поэтому в методе DCF для минимизации коллизий используется следующий алгоритм. После каждого успешного приема кадра принимающая сторона через короткий промежуток **SIFS (Short Interframe Space)** подтверждает успешный прием, посылая ответную квитанцию – **кадр ACK (ACKnowledgement)** (рис.15.10). Если в процессе передачи данных возникла коллизия, то передающая сторона не получает кадр ACK об успешном приеме. В этом случае размер CW-окна для передающего узла увеличивается почти вдвое. Так, если для первой передачи размер окна равен 31 слоту, то для второй попытки передачи он уже составляет 63 слота, для третьей – 127 слотов, для четвертой – 255, для пятой – 511, а для всех последующих – 1023 слота. То есть для каждой i-й передачи (если все предыдущие оказались безуспешными) размер CW-окна увеличивается по следующему правилу:

$$CW_i = 2CW_{i-1} + 1$$

Таким образом, увеличение размера окна происходит динамически по мере роста числа коллизий, что позволяет, с одной стороны, уменьшить временные задержки и, с другой стороны, снизить вероятность возникновения коллизий.

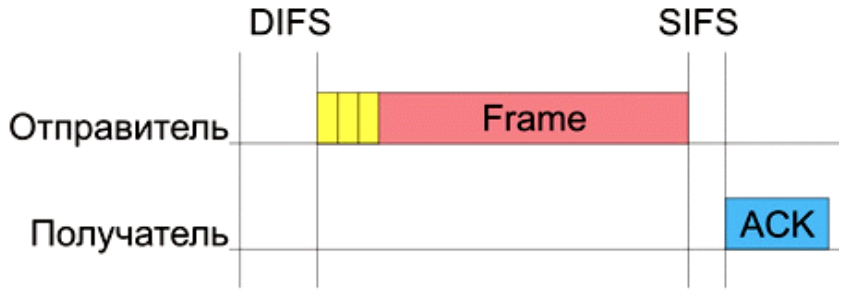


Рисунок 15.10. Кадры квитанции, отсылаемые в случае успешной передачи

Говоря об алгоритме реализации равноправного доступа к среде передачи данных, необходимо также учитывать и размер кадра данных. Действительно, если кадры данных будут слишком большими, то при возникновении коллизий придется повторно передавать большой объем информации, что приведет к снижению производительности сети. Кроме того, при большом размере кадров данных узлы сети вынуждены простаивать в течение довольно продолжительного времени, прежде чем начать передачу.

В то же время использование кадров данных небольшого размера, хотя и позволяет гарантировать равноправный доступ всех узлов к среде передачи данных и минимизирует издержки при возникновении коллизий, не может не отразиться негативно на полезном сетевом трафике. Дело в том, что каждый кадр наряду с полезной информацией содержит информацию служебную (заголовок кадра). При уменьшении размера кадра сокращается величина именно полезной информации (пользовательских данных), что обусловливает передачу по сети избыточного количества служебной информации. Поэтому размер кадра — это своего рода золотая середина, от правильного выбора которой зависит эффективность использования среды передачи данных.

Рассмотренный механизм регламентирования коллективного доступа к среде передачи данных имеет одно узкое место — так называемую проблему скрытых узлов. Из-за наличия естественных препятствий возможна ситуация, когда два узла сети не могут «слышать» друг друга напрямую. Такие узлы называют скрытыми.

Для того чтобы разрешить проблему скрытых узлов, функция DCF опционально предусматривает возможность использования алгоритма RTS/CTS.

Алгоритм RTS/CTS

В соответствии с алгоритмом RTS/CTS каждый узел сети, перед тем как послать данные в «эфир», сначала отправляет специальное короткое сообщение, которое называется **RTS (Ready To Send)** и означает готовность данного узла к отправке данных. Такое RTS-сообщение содержит информацию о продолжительности предстоящей передачи и об адресате и доступно всем узлам в

сети (если только они не скрыты от отправителя). Это позволяет другим узлам задержать передачу на время, равное объявленной длительности сообщения. Приемная станция, получив сигнал RTS, отвечает посылкой сигнала **CTS (Clear To Send)**, свидетельствующего о готовности станции к приему информации. После этого передающая станция посылает пакет данных, а приемная станция должна передать кадр ACK, подтверждающий безошибочный прием. Последовательность отправки кадров между двумя узлами сети показана на рис.17.10.

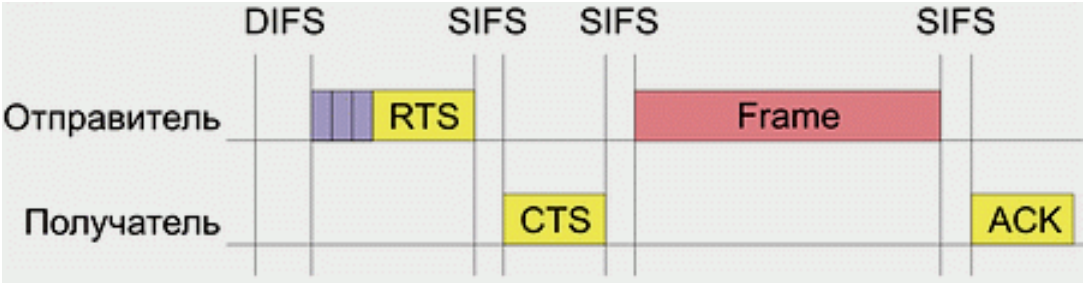


Рисунок 15.11. Взаимодействие между двумя узлами сети в соответствии с алгоритмом RTS/CTS

Теперь рассмотрим ситуацию, когда сеть состоит из четырех узлов: A, B, C и D (рис.15.12).

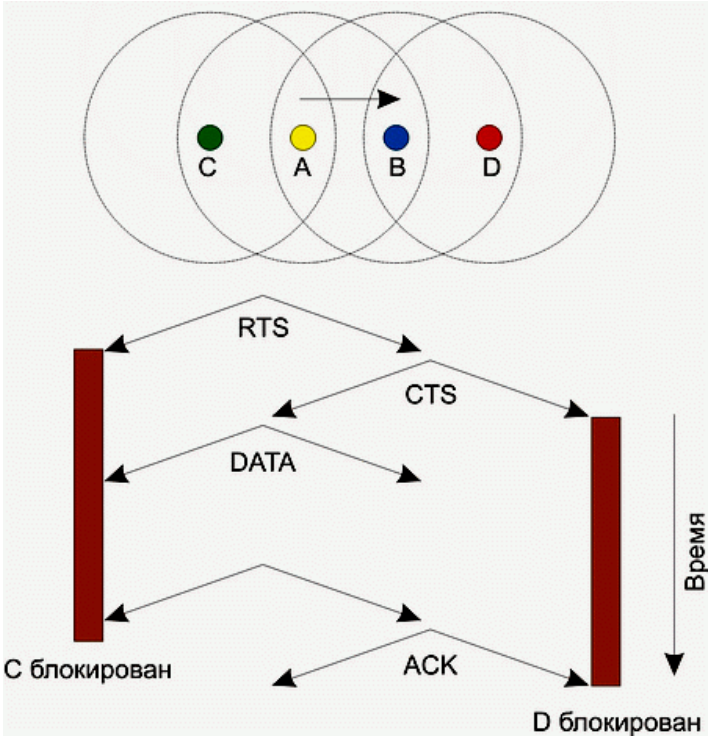


Рисунок 15.12. Решение проблемы скрытых узлов в алгоритме RTS/CTS.

Предположим, что узел C находится в зоне досягаемости только узла A, узел A находится в зоне досягаемости узлов C и B, узел B находится в зоне досягаемости узлов A и D, а узел D находится в зоне досягаемости только узла B. То есть в такой сети имеются скрытые узлы: узел C скрыт от узлов B и D, узел A скрыт от узла D.

В подобной сети алгоритм RTS/CTS позволяет справиться с проблемой возникновения коллизий, которая не решается посредством рассмотренного базового способа организации коллективного доступа в DCF. Действительно, пусть узел A пытается передать данные узлу B. Для этого он посылает сигнал RTS, который, помимо узла B, получает также узел C, но не получает узел D. Узел C, получив данный сигнал, блокируется, то есть приостанавливает попытки передавать сигнал до момента окончания передачи между узлами A и B. Узел B, в ответ на полученный сигнал RTS, посылает кадр CTS, который получают узлы A и D. Узел D, получив данный сигнал, также блокируется на время передачи между узлами A и B.

У алгоритма RTS/CTS имеются свои подводные камни, которые в определенных ситуациях могут приводить к снижению эффективности использования среды передачи данных. К примеру, в некоторых ситуациях, возможно такое явление, как распространение эффекта ложных блокировок узлов, что в конечном счете может привести к ступору в сети.

Функция централизованной координации PCF

Рассмотренный выше механизм распределенной координации DCF является базовым для протоколов 802.11 и может использоваться как в беспроводных сетях, функционирующих в режиме Ad-Hoc, так и в сетях, функционирующих в режиме Infrastructure, то есть в сетях, инфраструктура которых включает точку доступа.

Однако для сетей в режиме Infrastructure более естественным является несколько иной механизм регламентирования коллективного доступа, известный как функция централизованной координации **(Point Coordination Function, PCF)**.

Отметим, что механизм PCF является опциональным и применяется только в сетях с точкой доступа.

В случае задействования механизма PCF один из узлов сети (точка доступа) является центральным и называется **центром координации (Point Coordinator, PC)**. На центр координации возлагается задача управления коллективным доступом всех остальных узлов сети к среде передачи данных на основе определенного алгоритма опроса или исходя из приоритетов узлов сети. То есть центр координации опрашивает все узлы сети, внесенные в его список, и на основании этого опроса организует передачу данных между всеми узлами сети. Важно, что такой подход полностью исключает конкурирующий доступ к среде, как в случае механизма DCF, и делает невозможным возникновение коллизий, а для времезависимых приложений гарантирует приоритетный доступ к среде.

Функция централизованной координации не отрицает функцию распределенной координации, а скорее, дополняет ее, накладываясь поверх. Фактически в сетях с механизмом PCF реализуется как механизм PCF, так и традиционный механизм DCF. В течение определенного промежутка времени реализуется механизм PCF, затем – DCF, а потом все повторяется заново.

Для того чтобы иметь возможность чередовать режимы PCF и DCF, необходимо, чтобы точка доступа, выполняющая функции центра координации и реализующая режим PCF, имела бы приоритетный доступ к среде передачи данных. Это можно сделать, если использовать конкурентный доступ к среде передачи данных (как и в методе DCF), но для центра координации разрешить использовать промежуток ожидания, меньший DIFS. В этом случае если центр координации пытается получить доступ к среде, то он ожидает (как и все остальные узлы сети) окончания текущей передачи и, поскольку для него определяется минимальный режим ожидания после обнаружения «тишины» в эфире, первым получает доступ к среде. Промежуток ожидания, определяемый для центра координации, называется **PIFS (PCF Interframe Space)**, причем SIFS<PIFS<DIFS.

Режимы DCF и PCF объединяются в так называемом **суперфрейме**, который образуется из промежутка бесконкурентного доступа к среде, **называемого CFP (Contention-Free Period)**, и следующего за ним **промежутка конкурентного доступа к среде CP (Contention Period)** (рис.15.13).

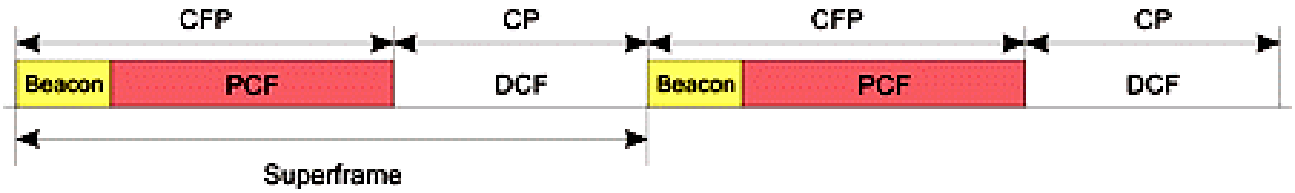


Рисунок 15.13. Объединение режимов PCF и DCF в одном суперфрейме

Суперфрейм начинается с **кадра-маячка (beacon)**, получив который все узлы сети приостанавливают попытки передавать данные на время, определяемое периодом CFP. Кадры маячки несут служебную информацию о продолжительности CFP-промежутка и позволяют синхронизировать работу всех узлов сети.

Во время режима PCF точка доступа опрашивает все узлы сети о кадрах, которые стоят в очереди на передачу, посылая им служебные кадры CF_POLL.

Опрашиваемые узлы в ответ на получение кадров CF_POLL посылают подтверждение CF_ACK. Если подтверждения не получено, то точка доступа переходит к опросу следующего узла.

Кроме того, чтобы иметь возможность организовать передачу данных между всеми узлами сети, точка доступа может передавать кадр данных (DATA) и совмещать кадр опроса с передачей данных (кадр DATA+CF_POLL). Аналогично узлы сети могут совмещать кадры подтверждения с передачей данных DATA+CF_ACK (рис. 15.14).

Допускаются следующие типы кадров во время режима PCF:

•DATA – кадр данных

•CF_ACK – кадр подтверждения

•CF_POLL – кадр опроса

•DATA+CF_ACK – комбинированный кадр данных и подтверждения

•DATA+CF_POLL – комбинированный кадр данных и опроса

•DATA+CF_ACK+CF_POLL - комбинированный кадр данных, подтверждения и опрос

• CF_ACK+CF_POLL – комбинированный кадр подтверждения и опроса

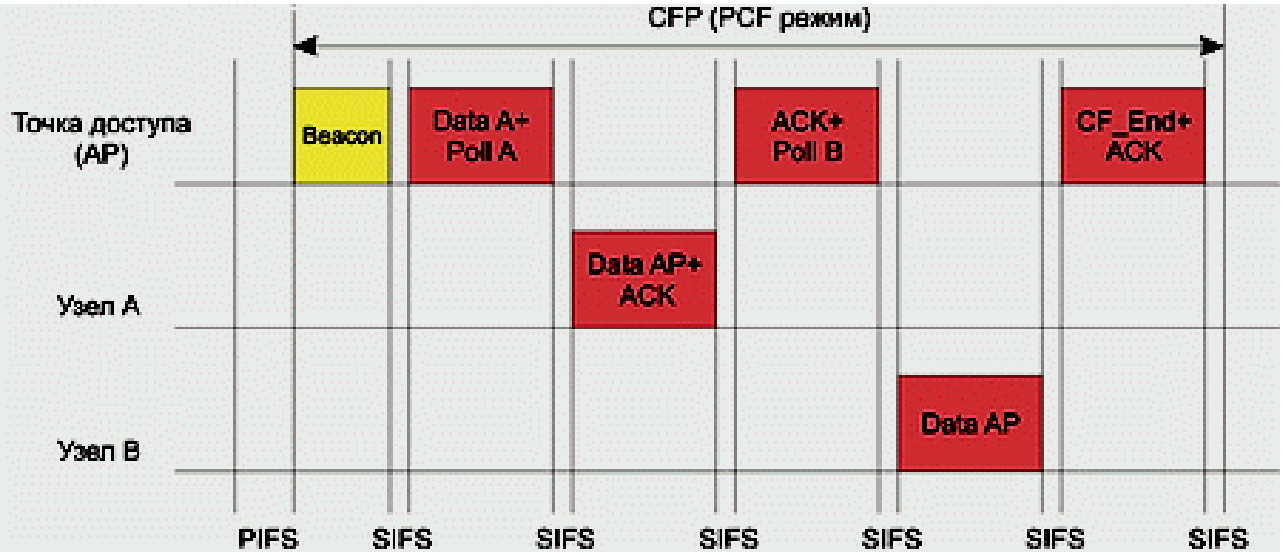


Рисунок 15.14. Организация передачи данных между узлами сети в режиме PCF

Максимальная длина кадра данных 802.11 равна 2346 байт, длина RTS-кадра — 20 байт, CTS-кадра — 14 байт. Так как RTS- и CTS-кадры гораздо короче, чем кадр данных, то потери данных в результате коллизии RTS- или CTS-кадров гораздо меньше, чем при коллизии кадров данных. Процедура обмена RTS- и CTS-кадрами не обязательна. От нее можно отказаться при небольшой нагрузке сети, поскольку в такой ситуации коллизии случаются редко, а значит, не стоит тратить дополнительное время на выполнение процедуры обмена RTS- и CTS-кадрами.

Весь обмен в сетях IEEE 802.11 происходит посредством отдельных кадров (frames). По их структуре особенно четко видно разделение на физический и MAC-уровни. Фактически кадр формируется на MAC-уровне, на физическом уровне к нему добавляется заголовок физического уровня (PLCP). На MAC-уровень пакеты передаются от приложений верхнего уровня. Если их размер превышает максимально допустимый в IEEE 802.11, происходит фрагментация —

большой пакет разбивается на несколько меньших, которые передаются по специальной процедуре.

Кадры MAC-уровня могут быть трех типов: кадры данных, контрольные (АСК, RTS, CTS и т. п.) и кадры управления (например, Beacon). Их структура одинакова (рис. 15.15).

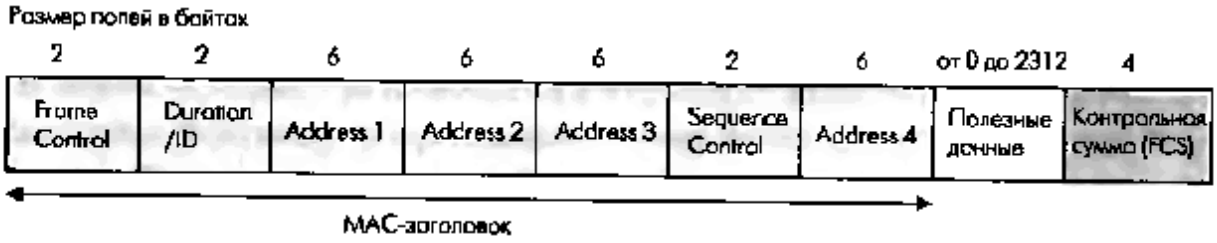


Рисунок 15.15. Структура кадров MAC-уровня сети 802.11

Каждый MAC-кадр включает MAC-заголовок, поле данных (Frame Body) и контрольную сумму CRC. В заголовке передается полная информация о версии протокола стандарта группы IEEE 802.11, типе кадра, системе защиты и т.д. (поле Frame Control); длительности процедуры передачи пакета (Duration/ID), адреса получателя/отправителя (Address l-4; четыре адресных поля необходимы, если пакеты передаются из подсети одной точки доступа в подсеть другой) и информация о последовательности связанных пакетов (Sequence Control). Поле данных может быть различной длины или вовсе отсутствовать (в контрольных кадрах).

16. СТРУКТУРИЗАЦИЯ ЛОКАЛЬНЫХ СЕТЕЙ

16.1. Причины структуризации локальных сетей

Первые локальные сети с небольшим (10-30) количеством компьютеров использовали только одну общую для всех подключенных к сети устройств разделяемую среду. При этом в соответствии с ограничениями технологий сети имели типовые топологии — общая шина (звезда) для Ethernet, кольцо для FDDI и Token Ring. Все перечисленные топологии обладают свойством однородности, то есть все компьютеры в такой сети неразличимы на уровне физических связей. Такая однородность структуры делает простой процедуру наращивания числа компьютеров, облегчает обслуживание и эксплуатацию сети.

Однако при построении больших сетей однородная структура связей превращается из достоинства в недостаток. В таких сетях использование типовых структур порождает различные ограничения, важнейшими из которых являются ограничения:

- на длину связи между узлами;
- на количество узлов в сети;
- на интенсивность трафика, порождаемого узлами сети.

Например, технология Ethernet на тонком коаксиальном кабеле позволяла использовать кабель длиной не более 185 метров, к которому можно было подключить не более 30 компьютеров. Однако если компьютеры начинали интенсивно обмениваться информацией между собой, тогда приходилось снижать число подключенных к кабелю компьютеров до 20, а то и до 10, чтобы каждому компьютеру доставалась приемлемая доля общей пропускной способности сети.

Для снятия этих ограничений стали использовать структуризацию сети на основе специального структурообразующего **коммуникационного оборудования,** в том числе повторителей, концентраторов, мостов, коммутаторов.

16.2. Физическая структуризация локальной сети

Различают топологию физических связей (физическую структуру сети) и топологию логических связей сети (логическую структуру сети).

В некоторых случаях физическая и логическая топологии сети совпадают. Например, сеть, представленная на рис.16.1***а***, имеет физическую кольцевую топологию. Пусть компьютеры этой сети используют метод детерминированного доступа. Причем токен всегда передается последовательно от компьютера к компьютеру в том же порядке, в котором компьютеры образуют физическое кольцо: то есть компьютер А передает токен компьютеру В, компьютер В — компьютеру С и т. д. В этом случае логическая топология сети также является кольцом.

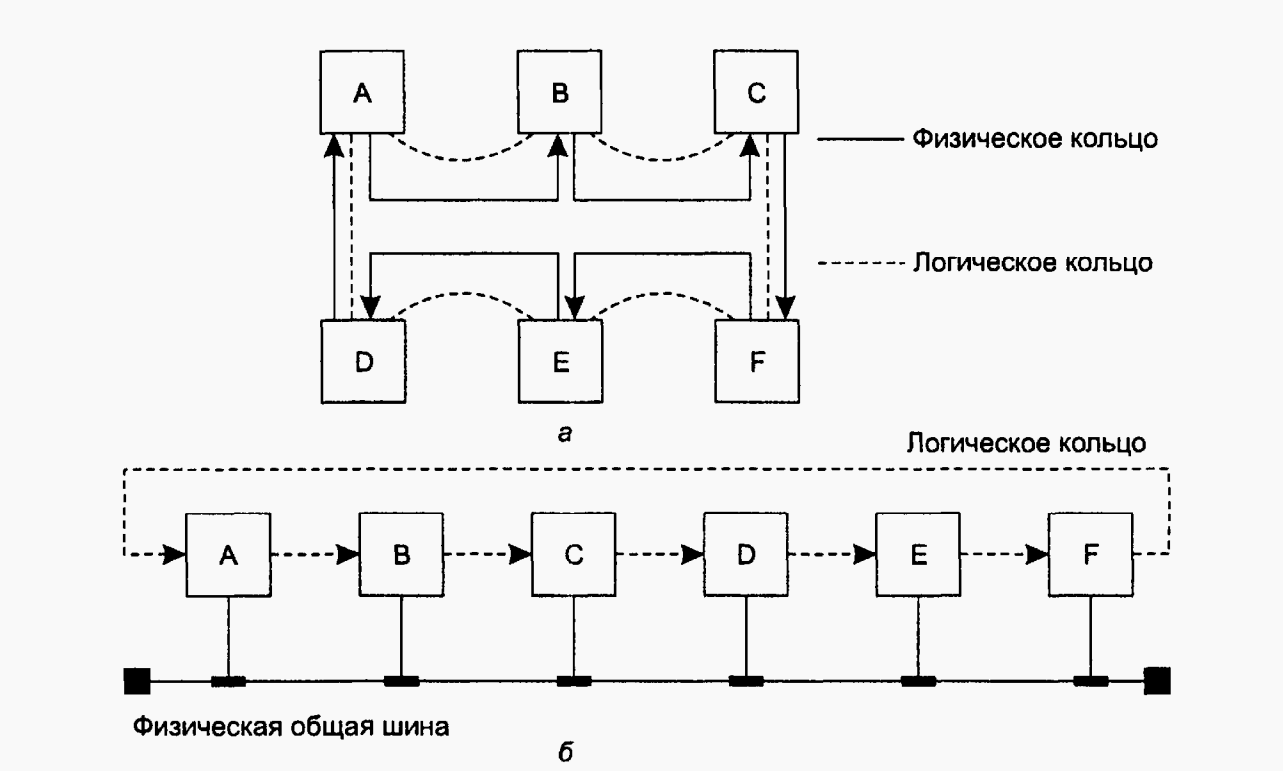


Рисунок 16.1. Физическая и логическая топологии.

Сеть, показанная на рис.16.1,***б***, являет собой пример несовпадения физической и логической топологий. Физически компьютеры соединены по топологии общая шина (звезда). Доступ же к шине происходит не по алгоритму случайного доступа, а путем передачи токена в кольцевом порядке: от компьютера А — компьютеру В, от компьютера В — компьютеру С и т. д. Здесь

порядок передачи токена уже не повторяет физические связи, а определяется логическим конфигурированием драйверов сетевых адаптеров. Ничто не мешает настроить сетевые адаптеры и их драйверы так, чтобы компьютеры образовали кольцо в другом порядке, например: B, A, C... При этом физическая структура сети никак не меняется.

Физическая структуризация единой разделяемой среды была первым шагом на пути построения более качественных локальных сетей. Цель физической структуризации — обеспечить построение сети не из одного, а из нескольких физических отрезков кабеля. Причем эти различные в физическом отношении отрезки должны были по-прежнему работать как единая разделяемая среда.

Простейшее из коммуникационных устройств — повторитель — используется для физического соединения различных сегментов кабеля локальной сети с целью увеличения общей длины сети. Повторитель повторяет сигналы, приходящие из одного сегмента сети в другие ее сегменты (рис. 16.2), улучшая их физические характеристики — мощность и форму сигналов, а также синхронность следования (исправляет неравномерность интервалов между импульсами). За счет этого повторитель позволяет преодолеть ограничения на длину линий связи. Так как поток сигналов, передаваемых узлом в сеть, распространяется по всем отрезкам сети, такая сеть остается сетью с единой разделяемой средой.

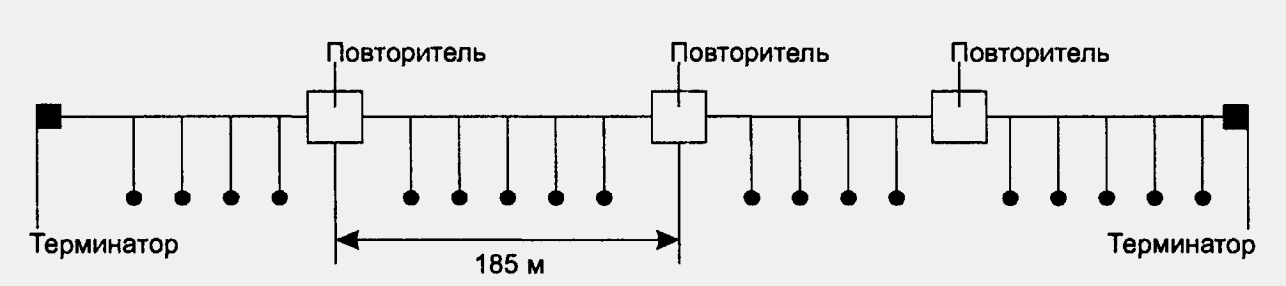


Рисунок 16.2. Повторители позволяют увеличить длину сети

Повторитель, который имеет несколько портов и соединяет несколько физических сегментов, часто называют концентратором, или хабом. Эти названия

отражают тот факт, что в данном устройстве сосредоточиваются все связи между сегментами сети.

Добавление в сеть повторителя всегда изменяет ее физическую топологию, но при этом оставляет без изменения логическую топологию.

Концентраторы являются необходимыми устройствами практически во всех базовых технологиях локальных сетей — Ethernet, ArcNet, Token Ring, FDDI, Fast Ethernet, Gigabit Ethernet, lOOVG-AnyLAN. В работе концентраторов любых технологий много общего — они повторяют сигналы, пришедшие с одного из своих портов, на других своих портах. Разница состоит в том, на каких именно портах повторяются входные сигналы. Так, концентратор Ethernet повторяет входной сигнал на *всех* своих портах, кроме того, с которого этот сигнал поступил (рис. 16.3, *а*). А концентратор Token Ring (рис. 16.3, *б*) повторяет входной сигнал только на *одном*, *соседнем* порту.

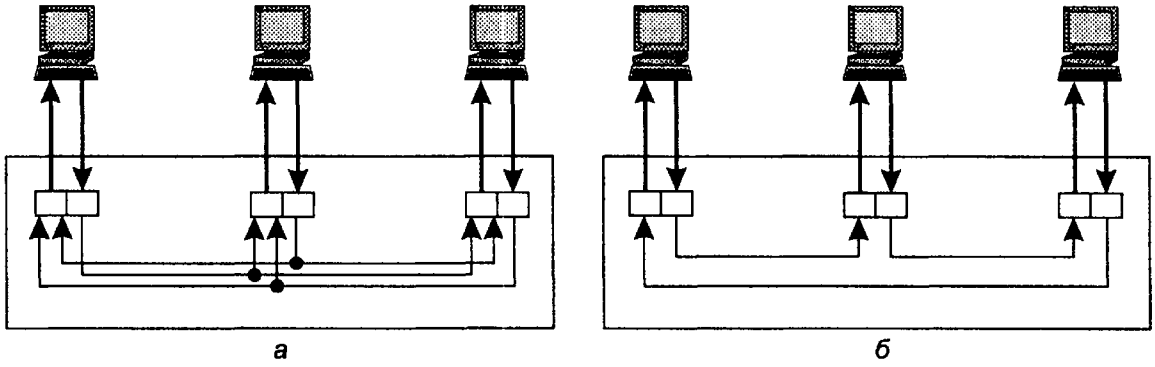


Рисунок 16.3. Концентраторы различных технологий

16.3.Логическая структуризация сети на разделяемой среде

Физическая структуризация сети не позволяет справиться с такими важными проблемами, как дефицит пропускной способности, невозможность использования в разных частях сети линий связи разной пропускной способности. В таком случае может помочь логическая структуризация сети.

Типовые физические топологии сети (шина, кольцо, звезда), которые ограничивают все сетевые устройства, предоставляя им для обмена данными только одну разделяемую среду, оказываются неадекватными структуре

информационных потоков в большой сети. Например, в сети с общей шиной взаимодействие любой пары компьютеров занимает ее на все время обмена, поэтому при увеличении числа компьютеров в сети шина становится узким местом.

При построении небольших сетей, состоящих из 10-30 узлов, использование стандартных технологий на разделяемой среде приводит к экономичным и эффективным решениям, что проявляется в первую очередь в следующих свойствах:

- простая топология сети допускает легкое наращивание числа узлов (в небольших пределах);

- отсутствуют потери кадров из-за переполнения буферов коммуникационных устройств, так как сам метод доступа к разделяемой среде регулирует поток кадров и приостанавливает станции, слишком часто генерирующие кадры;

- простота протоколов обеспечивает низкую стоимость сетевых адаптеров, повторителей и концентраторов и сети в целом.

Однако справедливым является и другое утверждение - крупные сети, насчитывающие сотни и тысячи узлов, не могут быть построены на основе одной разделяемой среды даже при такой скоростной технологии, как Gigabit Ethernet. И не только потому, что проектировщик сети часто сталкивается с жесткими ограничениями максимальной длины сети, обусловленными особенностями метода доступа Ethernet. И не только потому, что практически все технологии ограничивают количество узлов в разделяемой среде: все технологии семейства Ethernet — 1024 узлами, Token Ring - 260 узлами, а FDDI - 500 узлами.

Главная проблема сетей с разделяемой средой – дефицит пропускной способности.

На рис. 16.4 показаны зависимости задержек доступа к общей среде передачи от коэффициента использования среды, полученные для сетей Ethernet, Token Ring и FDDI путем имитационного моделирования.

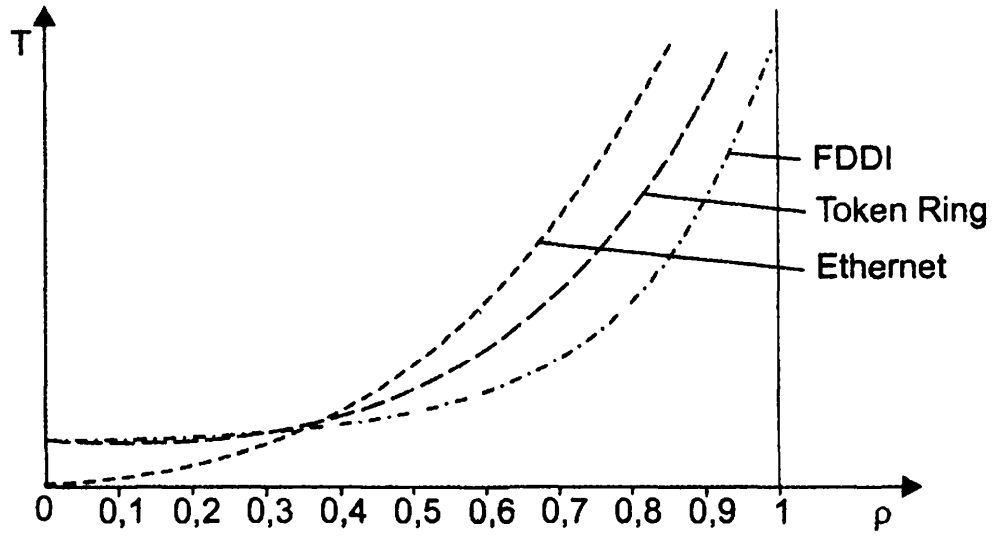


Рисунок 16.4. Задержки доступа к среде передачи данных для технологий Ethernet, Token Ring и FDDI

Как видно из рисунка, всем технологиям присуща качественно одинаковая картина экспоненциального роста величины задержек доступа при увеличении коэффициента использования сети. Однако их отличает порог, при котором наступает резкий перелом в поведении сети, когда почти прямолинейная зависимость переходит в крутую экспоненциальную. Для всего семейства технологий Ethernet — это 30-50 % (сказывается эффект коллизий), для технологии Token Ring — 60 %, а технологии FDDI - 70-80 %.

Количество узлов, при которых коэффициент использования сети начинает приближаться к опасной границе, зависит от типа функционирующих в узлах приложений. Если раньше для сетей Ethernet считалось, что 30 узлов — это вполне приемлемое число для одного разделяемого сегмента, то сегодня, в условиях, когда мультимедийные приложения передают по сети большие файлы данных, предельное число узлов может составлять 5-10.

Преимущества логической структуризации сети

Ограничения, возникающие из-за использований одной разделяемой среды, можно преодолеть, выполнив ***логическую структуризацию сети,*** то есть сегментировать единую разделяемую среду на несколько и: соединить

полученные сегменты сети такими устройствами, как мосты, коммутаторы или маршрутизаторы.

Перечисленные устройства передают кадры с одного своего порта на другой, анализируя адрес назначения, помещенный в этих кадрах. Мосты и коммутаторы выполняют операцию передачи кадров на основе плоских адресов канального уровня, то есть MAC-адресов, а маршрутизаторы используют для этой цели иерархические адреса сетевого уровня.

Логическая структуризация позволяет решить несколько задач, основные из них: повышение производительности, гибкости, безопасности и управляемости сети.

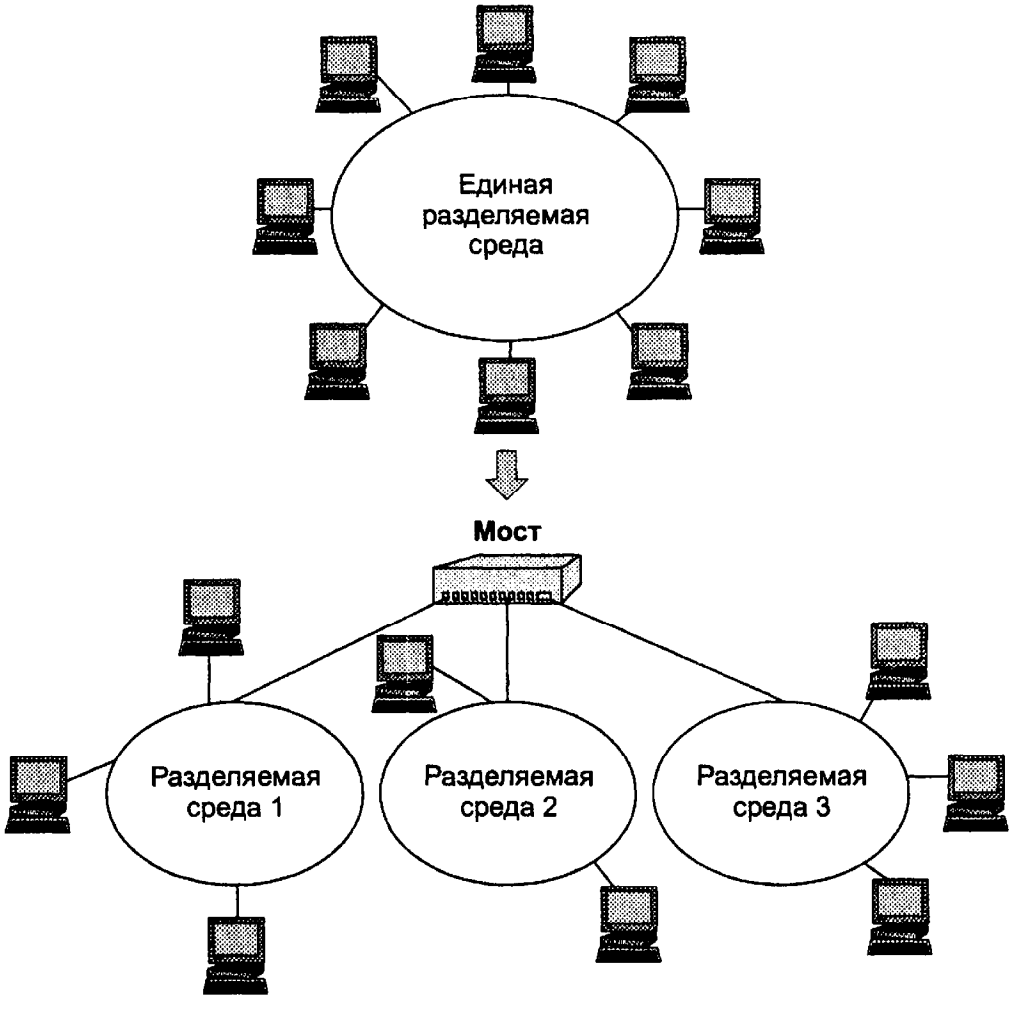


Рисунок 16.5. Логическая структуризация сети

Повышение производительности. Для иллюстрации эффекта повышения производительности, который является главной целью логической структуризации, рассмотрим рис. 16.6. На нем показаны два сегмента Ethernet, соединенные мостом. Внутри сегментов имеются повторители. До деления сети на сегменты весь трафик, генерируемый узлами сети, был общим (представим, что вместо моста был повторитель) и учитывался при определении коэффициента использования сети. Если обозначить среднюю интенсивность трафика, идущего от узла i к узлу j, через C_{ij}, то суммарный трафик, который должна была передавать сеть до деления на сегменты, равен $C_{\Sigma} = \sum C_{ij}$ (считаем, что суммирование проводится по всем узлам).

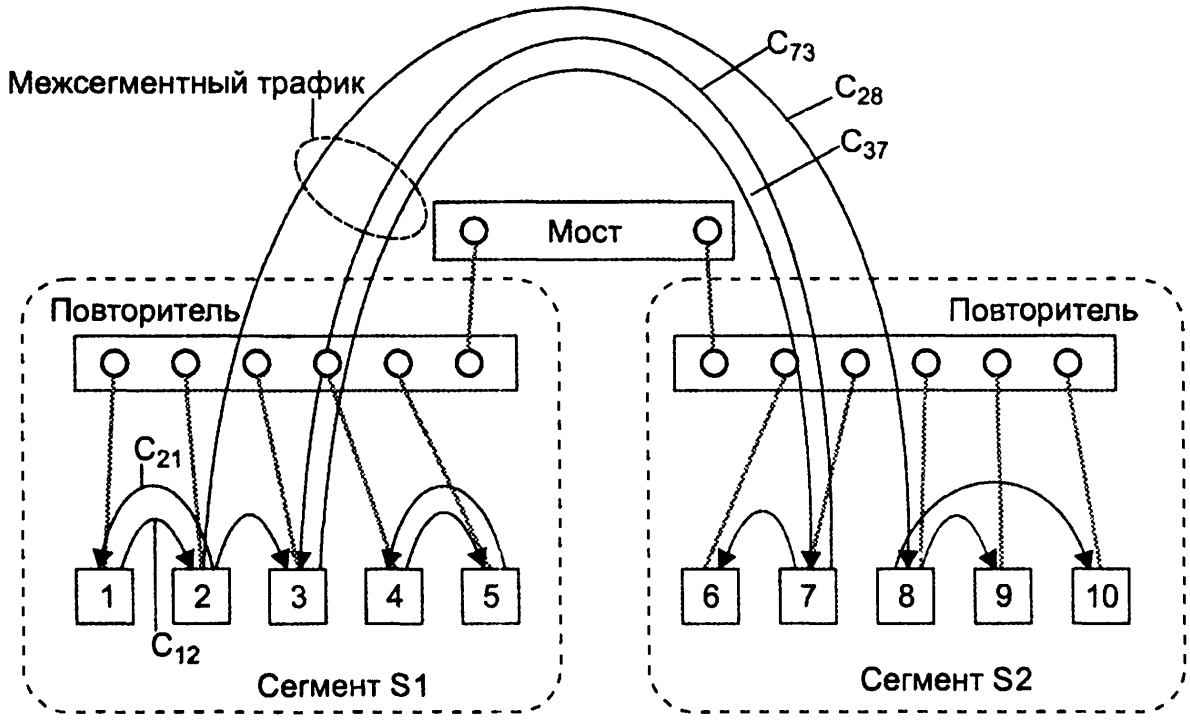


Рисунок 16.6. Изменение нагрузки при делении сети на сегменты

После разделения сети подсчитаем нагрузку отдельно для каждого сегмента. Например, нагрузка сегмента S1 стала равна $C_{S1} + C_{S1-S2}$ где C_{S1} — внутренний трафик сегмента S1, а $C_{S1\text{-}S2}$ — межсегментный трафик. Чтобы показать, что нагрузка сегмента S1 стала меньше, чем нагрузка исходной сети,

заметим, что общую нагрузку сети до разделения на сегменты можно представить в таком виде:

$$C_{\Sigma} = C_{S1} + C_{S1-S2} + C_{S2}$$

Значит, нагрузка сегмента S1 после разделения стала равной $C_{\Sigma} - C_{S2}$, то есть стала меньше на величину внутреннего трафика сегмента S2. Аналогичные рассуждения можно повторить относительно сегмента S2. Следовательно, в соответствии с графиками, приведенными на рис.16.4, задержки в сегментах уменьшились, а полезная пропускная способность, приходящаяся на один узел, увеличилась.

На практике в сети всегда можно выделить группу компьютеров, которые принадлежат сотрудникам, решающим общую задачу. Это могут быть сотрудники одной рабочей группы, отдела, другого структурного подразделения предприятия. В большинстве случаев им нужен доступ к ресурсам сети их отдела и только изредка — доступ к удаленным ресурсам.

В 80-е годы существовало эмпирическое правило, говорящее о том, что можно разделить сеть на сегменты так, что 80 % трафика составят обращения к локальным ресурсам и только 20 % — к удаленным. Сегодня такая закономерность не всегда соответствует действительности, она может трансформироваться в правило 50 на 50% и даже 20 на 80% (например, большая часть обращений направлена к ресурсам Интернета или к централизованным серверам предприятия). Тем не менее, в любом случае внутрисегментный трафик существует. Если его нет, значит, **сеть разбита на логические сегменты неверно**.

Повышение гибкости сети. При построении сети как совокупности сегментов каждый из них может быть адаптирован к специфическим потребностям рабочей группы или отдела. Например, в одном сегменте может использоваться технология Ethernet и ОС Unix, в другом — Token Ring и OS-400. Вместе с тем, у пользователей обоих сегментов есть возможность обмениваться данными через мосты/ коммутаторы. Процесс разбиения сети на логические

сегменты можно рассматривать и в обратном направлении, как процесс создания большой сети из уже имеющихся небольших сетей.

Повышение безопасности данных. Устанавливая различные логические фильтры на мостах/коммутаторах, можно контролировать доступ пользователей к ресурсам других сегментов, чего не позволяют делать повторители.

Повышение управляемости сети. Побочным эффектом снижения трафика и повышения безопасности данных является упрощение управления сетью. Проблемы очень часто локализуются внутри сегмента. Сегменты образуют логические домены управления сетью.

Мост (bridge) делит единую среду передачи на части (часто называемые **логическими сегментами),** передавая информацию из одного сегмента в другой только в том случае, если такая передача действительно необходима, то есть если адрес компьютера назначения принадлежит другому сегменту. Тем самым мост изолирует трафик одного сегмента от трафика другого, повышая общую производительность сети.

Коммутатор (switch) функционально подобен мосту и отличается от моста в основном более высокой производительностью. Каждый интерфейс коммутатора оснащен специализированным процессором, который обрабатывает кадры по алгоритму моста независимо от процессоров других портов. За счет этого общая производительность коммутатора обычно намного выше производительности традиционного моста, имеющего один процессорный блок. Можно сказать, что коммутаторы — это усовершенствованные мосты, которые обрабатывают кадры в параллельном режиме. Когда стало экономически оправданно использовать отдельные специализированные процессоры на каждом порту коммуникационного устройства, коммутаторы локальных сетей полностью вытеснили мосты.

Оба эти устройства продвигают кадры на основании одного и того же алгоритма, а именно **алгоритма прозрачного моста,** описанного в стандарте IEEE 802.1D.

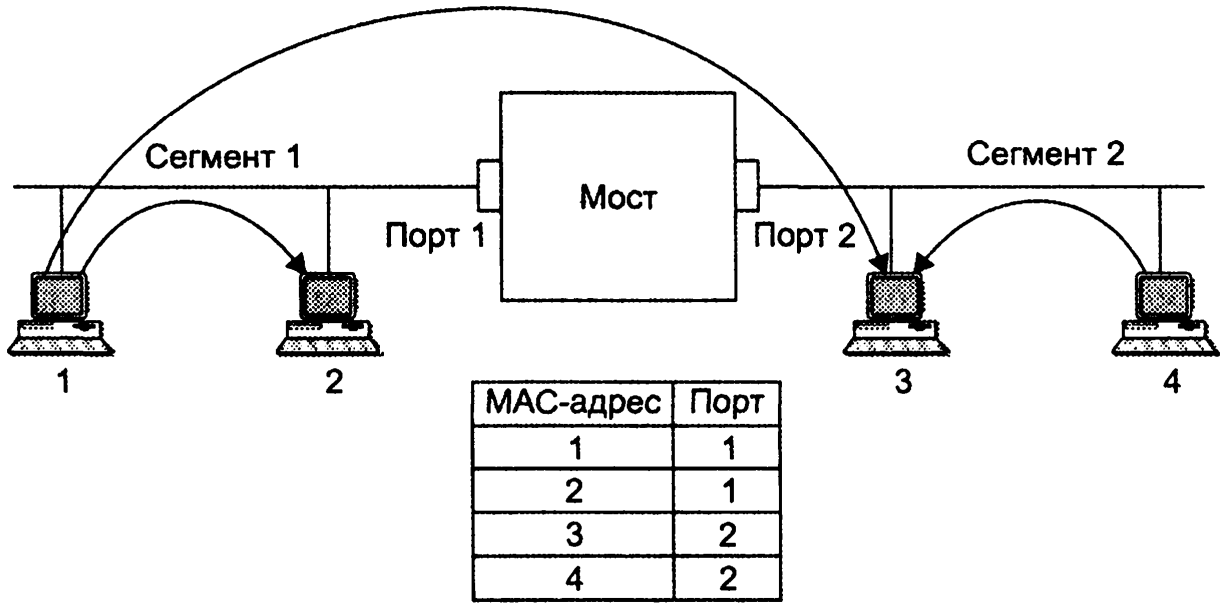
16.4. Алгоритм прозрачного моста IEEE 802.1D

Слово «прозрачный» в названии ***алгоритм прозрачного моста*** отражает тот факт, что мосты и коммутаторы в своей работе не учитывают существование в сети сетевых адаптеров конечных узлов, концентраторов, повторителей. С другой стороны, и перечисленные выше сетевые устройства функционируют, «не замечая» присутствия в сети мостов и коммутаторов.

Алгоритм прозрачного моста не зависит от технологии локальной сети, в которой устанавливается мост/коммутатор, поэтому прозрачные мосты/коммутаторы Ethernet работают точно так же, как прозрачные мосты/коммутаторы FDDI или Token Ring.

Коммутатор строит свою адресную таблицу на основании пассивного наблюдения за трафиком, циркулирующим в подключенных к его портам сегментах. При этом коммутатор учитывает адреса источников кадров данных, поступающих на порты коммутатора. По адресу источника кадра коммутатор делает вывод о принадлежности узла-источника тому или иному сегменту сети.

Рассмотрим процесс автоматического создания адресной таблицы коммутатора и ее использования на примере простой сети, представленной на рис.16.7.



MAC-адрес	Порт
1	1
2	1
3	2
4	2

Рисунок 16.7. Принцип работы прозрачного моста/коммутатора

Коммутатор соединяет два сетевых сегмента. Сегмент 1 составляют компьютеры, подключенные с помощью одного отрезка коаксиального кабеля к порту 1 коммутатора, а сегмент 2 — компьютеры, подключенные с помощью другого отрезка коаксиального кабеля к порту 2 коммутатора. В исходном состоянии коммутатор не знает о том, компьютеры с какими MAC-адресами подключены к каждому из его портов. В этой ситуации коммутатор просто передает любой захваченный и буферизованный кадр на ***все*** свои порты за исключением того порта, от которого этот кадр получен. В нашем примере у коммутатора только два порта, поэтому он передает кадры с порта 1 на порт 2, и наоборот. Отличие работы коммутатора в этом режиме от повторителя заключается в том, что он передает кадр, предварительно буферизуя его, а не бит за битом, как это делает повторитель. Буферизация разрывает логику работы всех сегментов как единой разделяемой среды. Когда коммутатор собирается передать кадр с сегмента на сегмент, например с сегмента 1 на сегмент 2, он, как обычный конечный узел, пытается получить доступ к разделяемой среде сегмента 2 по правилам алгоритма доступа, в данном примере — по правилам алгоритма CSMA/CD.

Одновременно с передачей кадра на все порты коммутатор изучает адрес источника кадра и делает запись о его принадлежности к тому или иному сегменту в своей **адресной таблице.** Эту таблицу также называют **таблицей фильтрации,** или **таблицей маршрутизации.** Например, получив на порт 1 кадр от компьютера 1, коммутатор делает первую запись в своей адресной таблице.

MAC-адрес 1 — порт 1.

Эта запись означает, что компьютер, имеющий MAC-адрес 1, принадлежит сегменту, подключенному к порту 1 коммутатора. Если все четыре компьютера данной сети проявляют активность и посылают друг другу кадры, то скоро коммутатор построит полную адресную таблицу сети, состоящую из 4 записей — по одной записи на узел (см. рис. 16.7).

При каждом поступлении кадра на порт коммутатора он, прежде всего, пытается найти адрес назначения кадра в адресной таблице. Продолжим рассмотрение действий коммутатора на примере (см. рис. 16.7).

1. При получении кадра, направленного от компьютера 1 компьютеру 3, коммутатор просматривает адресную таблицу на предмет совпадения адреса в какой-либо из ее записей с адресом назначения — MAC-адресом 3. Запись с искомым адресом имеется в адресной таблице.

2. Коммутатор выполняет второй этап анализа таблицы — проверяет, находятся ли компьютеры с адресами источника и назначения в одном сегменте. В примере компьютер 1 (MAC-адрес 1) и компьютер 3 (MAC-адрес 3) находятся в разных сегментах. Следовательно, коммутатор выполняет операцию **продвижения** (forwarding) **кадра** — передает кадр на порт 2, который подключен к сегменту получателя, получает доступ к сегменту и передает туда кадр.

3. Если бы оказалось, что компьютеры принадлежали одному сегменту, то кадр просто удаляется из буфера. Такая операция называется **фильтрацией** (filtering).

4. Если бы запись MAC-адрес 3 отсутствовала в адресной таблице, то есть, другими словами, *адрес назначения был неизвестен* коммутатору, то он передал бы кадр на все свои порты, кроме порта — источника кадра, как и на начальной стадии процесса обучения.

Процесс обучения коммутатора никогда не заканчивается и происходит одновременно с продвижением и фильтрацией кадров. Коммутатор постоянно следит за адресами источника буферизуемых кадров, чтобы автоматически приспосабливаться к изменениям, происходящим в сети, — перемещениям компьютеров из одного сегмента сети в другой, отключению и появлению новых компьютеров.

Входы адресной таблицы могут быть динамическими, создаваемыми в процессе самообучения коммутатора, и статическими, создаваемыми вручную администратором сети. **Статические записи** не имеют срока жизни, что дает администратору возможность влиять на работу коммутатора, например,

ограничивая передачу кадров с определенными адресами из одного сегмента в другой.

Динамические записи имеют срок жизни — при создании или обновлении записи в адресной таблице с ней связывается отметка времени. По истечении определенного тайм-аута запись помечается как недействительная, если за это время коммутатор не принял ни одного кадра с данным адресом в поле адреса источника. Это дает возможность коммутатору автоматически реагировать на перемещения компьютера из сегмента в сегмент — при его отключении от старого сегмента запись о его принадлежности к нему со временем вычеркивается из адресной таблицы.

Кадры с широковещательными MAC-адресами, как и кадры с неизвестными адресами назначения, передаются коммутатором на все его порты. Такой режим распространения кадров называется **затоплением сети** (flooding). Наличие коммутаторов в сети не препятствует распространению широковещательных кадров по всем сегментам сети. Однако это является достоинством только тогда, когда широковещательный адрес выработан корректно работающим узлом.

Нередко в результате каких-либо программных или аппаратных сбоев протокол верхнего уровня или сетевой адаптер начинают работать некорректно, а именно постоянно с высокой интенсивностью генерировать кадры с широковещательным адресом. Коммутатор в соответствии со своим алгоритмом передает ошибочный трафик во все сегменты. Такая ситуация называется **широковещательным штормом** (broadcast storm).

К сожалению, коммутаторы не защищают сети от широковещательного шторма, во всяком случае, по умолчанию, как это делают маршрутизаторы. Максимум, что может сделать администратор с помощью коммутатора для борьбы с широковещательным штормом, — установить для каждого узла предельно допустимую интенсивность генерации кадров с широковещательным адресом. Но при этом нужно точно знать, какая интенсивность является нормальной, а какая — ошибочной. При смене протоколов ситуация в сети может

измениться, и то, что вчера считалось ошибочным, сегодня может оказаться нормой.

Протокол, реализующий алгоритм коммутатора, располагается между уровнями MAC и LLC (рис. 16.8).

Рисунок 16.8. Место протокола коммутатора в стеке протоколов

16.5. Топологические ограничения коммутаторов в локальных сетях

Рассмотрим это ограничение на примере сети, показанной на рис. 16.9.

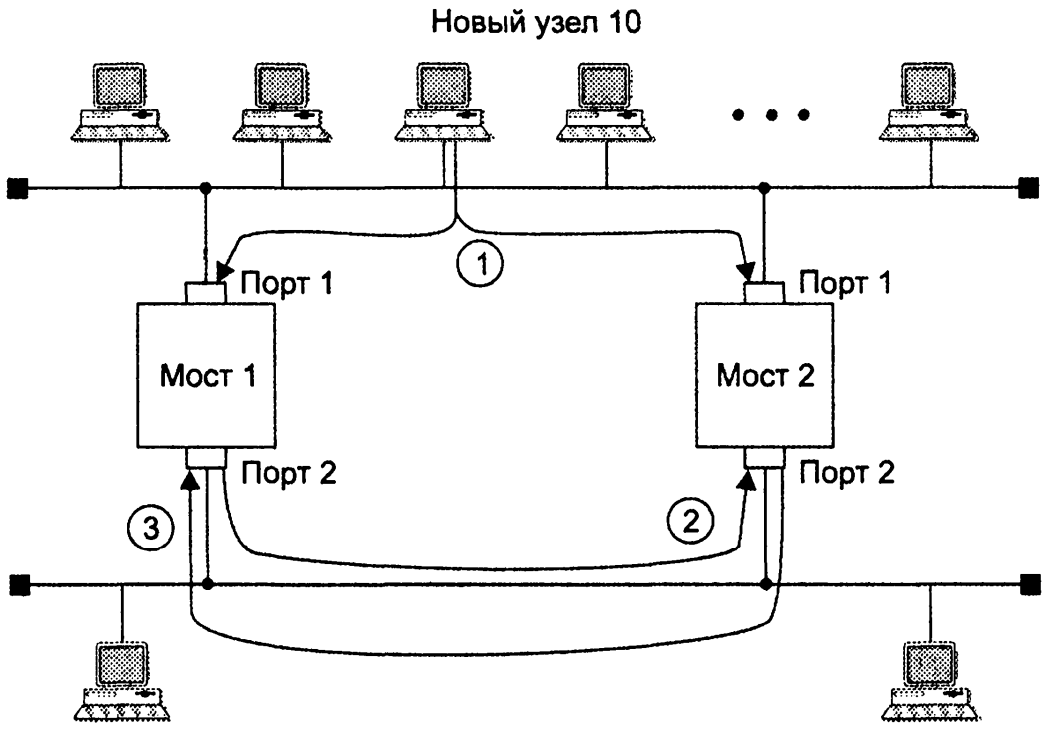


Рисунок 16.9. Влияние замкнутых контуров на работу коммутатора

Два сегмента Ethernet параллельно соединены двумя коммутаторами, так что образовалась петля. Пусть новая станция с MAC-адресом 10 впервые начинает работу в данной сети. Обычно начало работы любой операционной системы сопровождается рассылкой широковещательных кадров, в которых станция заявляет о своем существовании и одновременно ищет серверы сети.

На этапе 1 станция посылает первый кадр с широковещательным адресом назначения и адресом источника 10 в свой сегмент. Кадр попадает как в коммутатор 1, так и в коммутатор 2. В обоих коммутаторах новый адрес источника 10 заносится в адресную таблицу с пометкой о его принадлежности сегменту 1, то есть создается новая запись вида:

MAC-адрес 10 - Порт 1.

Так как адрес назначения широковещательный, то каждый коммутатор должен передать кадр на сегмент 2. Эта передача происходит поочередно в соответствии с методом случайного доступа технологии Ethernet. Пусть первым доступ к сегменту 2 получает коммутатор 1 (этап 2 на рис. 12). При появлении кадра на сегменте 2 коммутатор 2 принимает его в свой буфер и обрабатывает. Он видит, что адрес 10 уже есть в его адресной таблице, но пришедший кадр является более свежим, и он решает, что адрес 10 принадлежит сегменту 2, а не 1. Поэтому коммутатор 2 корректирует содержимое базы и делает запись о том, что адрес 10 принадлежит сегменту 2:

MAC-адрес 10 - Порт 2.

Аналогично поступает коммутатор 1, когда коммутатор 2 передает свою копию кадра на сегмент 2.

Ниже перечислены последствия наличия петли в сети.

1. «Размножение» кадра, то есть появление нескольких его копий (в данном случае — двух, но если бы сегменты были соединены тремя коммутаторами — то трех и т. д.).

2. Бесконечная циркуляция обеих копий кадра по петле в противоположных направлениях, а значит, засорение сети ненужным трафиком.

3. Постоянная перестройка коммутаторами своих адресных таблиц, так как кадр с адресом источника 10 будет появляться то на одном порту, то на другом.

В целях исключения всех этих нежелательных эффектов коммутаторы нужно применять так, чтобы между логическими сегментами не было петель, то есть строить с помощью коммутаторов только древовидные структуры, гарантирующие наличие единственного пути между любыми двумя сегментами. Тогда кадры от каждой станции будут поступать в коммутатор всегда с одного и того же порта, и коммутатор сможет правильно решать задачу выбора рационального маршрута в сети.

В небольших сетях сравнительно легко гарантировать существование одного и только одного пути между двумя сегментами. Но когда количество соединений возрастает, то вероятность непреднамеренного образования петли оказывается высокой.

Возможна и другая причина возникновения петель. Так, для повышения надежности желательно иметь между коммутаторами резервные связи, которые не участвуют в нормальной работе основных связей по передаче информационных кадров станций, но при отказе какой-либо основной связи образуют новую связную рабочую конфигурацию без петель.

Избыточные связи необходимо блокировать, то есть переводить их в неактивное состояние. В сетях с простой топологией эта задача решается вручную, путем блокирования соответствующих портов коммутаторов. В больших сетях со сложными связями используются алгоритмы, которые позволяют решать задачу обнаружения петель автоматически. Наиболее известным из них является стандартный **алгоритм покрывающего дерева** (Spanning Tree Algorithm, STA).

17. ДУПЛЕКСНЫЕ ПРОТОКОЛЫ ЛОКАЛЬНЫХ СЕТЕЙ

17.1. Изменения в работе MAC-уровня в дуплексном режиме

Технология коммутации сама по себе не имеет непосредственного отношения к методу доступа к среде, который используется портами коммутатора. При подключении к порту коммутатора сегмента, представляющего собой разделяемую среду, данный порт, как и все остальные узлы такого сегмента, должен поддерживать полудуплексный режим.

Однако когда к каждому порту коммутатора подключен не сегмент, а только *один* компьютер, причем по двум физически раздельным каналам, как это происходит почти во всех стандартах Ethernet, кроме коаксиальных версий Ethernet, ситуация становится не такой однозначной. Порт может работать как в обычном полудуплексном режиме, так и в дуплексном.

Подключение к портам коммутаторов не сегментов, а отдельных компьютеров называется ***микросегментацией.***

В обычном для Ethernet **полудуплексном режиме** работы порт коммутатора по-прежнему распознает коллизии. Доменом коллизий в этом случае является участок сети, включающий передатчик коммутатора, приемник коммутатора, передатчик сетевого адаптера компьютера, приемник сетевого адаптера компьютера и две витые пары, соединяющие передатчики с приемниками (рис. 17.1).

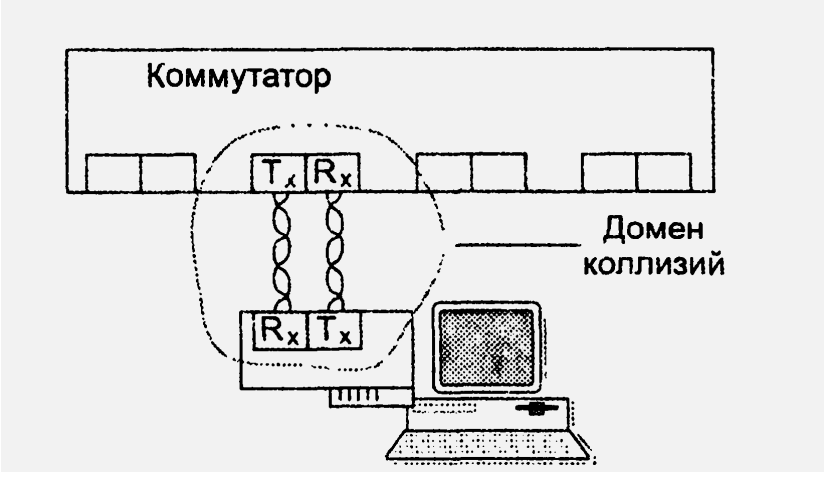


Рисунок 17.1. Домен коллизий, образуемый компьютером и портом коммутатора

Коллизия возникает, когда передатчики порта коммутатора и сетевого адаптера одновременно или почти одновременно начинают передачу своих кадров, считая, что сегмент свободен. Правда, вероятность коллизии в таком сегменте гораздо меньше, чем в сегменте, состоящем из 20-30 узлов, но она не нулевая. При этом максимальная производительность сегмента Ethernet в 14 880 кадров в секунду при минимальной длине кадра делится между передатчиком порта коммутатора и передатчиком сетевого адаптера. Если считать, что она делится пополам, то каждому предоставляется возможность передавать примерно по 7440 кадров в секунду.

В дуплексном режиме одновременная передача данных передатчиком порта коммутатора и сетевого адаптера коллизией не считается. В принципе, это достаточно естественный режим работы для отдельных дуплексных каналов передачи данных, и он часто используется в протоколах глобальных сетей. При дуплексной связи порты Ethernet 10 Мбит/с могут передавать данные со скоростью 20 Мбит/с — по 10 Мбит/с в каждом направлении.

Естественно, необходимо, чтобы MAC-узлы взаимодействующих устройств поддерживали дуплексный режим. В случае, когда только один узел поддерживает дуплексный режим, второй узел будет постоянно фиксировать коллизии и приостанавливать свою работу, в то время как другой узел продолжит передавать данные, которые никто в этот момент не принимает. Изменения, которые нужно внести в логику работы MAC-узла, чтобы он мог работать в дуплексном режиме, минимальны — нужно просто отменить фиксацию и обработку коллизий в сетях Ethernet. Если же микросегмент образован компьютером, поддерживающим протокол Token Ring или FDDI, то сетевой адаптер и порт коммутатора должны посылать свои кадры, не дожидаясь прихода токена доступа, а тогда, когда в этом возникнет необходимость. Фактически, при работе в дуплексном режиме MAC-узел игнорирует метод доступа к среде, разработанный для данной технологии.

17.2.Борьба с перегрузками

В классическом полудуплексном режиме у коммутатора имеется возможность воздействовать на конечный узел с помощью алгоритма доступа к среде, который соседний узел обязан отрабатывать. Применяются два основных способа управления потоком кадров — обратное давление на конечный узел и агрессивный захват среды.

Метод обратного давления (backpressure) состоит в создании искусственных коллизий в сегменте, который чересчур интенсивно посылает кадры в коммутатор. Для этого коммутатор обычно использует jam-последовательность, отправляемую на выход порта, к которому подключен сегмент (или узел), чтобы приостановить его активность.

Другой метод «торможения» обычно применяется в том случае, когда соседом является конечный узел. Метод основан на так называемом **агрессивном захвате среды** либо после окончания передачи очередного кадра, либо после коллизии. Эти два случая иллюстрирует рис. 17.2.

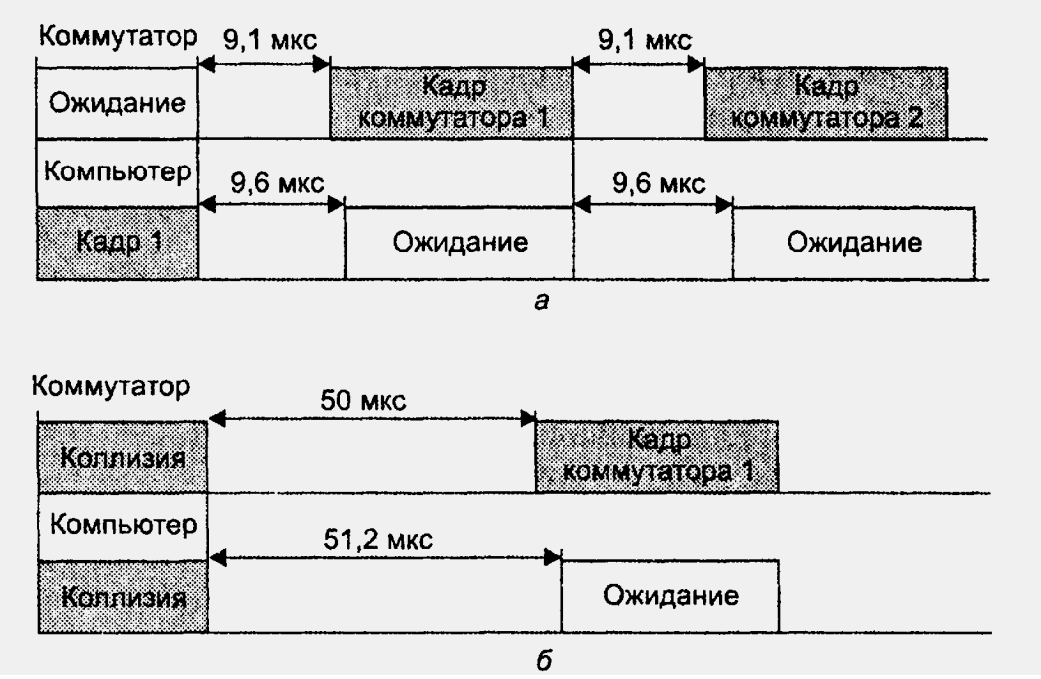


Рисунок 17.2. Агрессивное поведение порта коммутатора при перегрузках

В первом случае (рис. 17.2, *а*) коммутатор окончил передачу очередного кадра и вместо технологической паузы в 9,6 мкс сделал паузу в 9,1 мкс после

чего начал передачу нового кадра. Компьютер не смог захватить среду, так как он выдержал стандартную паузу в 9,6 мкс и обнаружил после этого, что среда уже занята.

Во втором случае (рис. 17.2, *б*) кадры коммутатора и компьютера столкнулись, то есть была зафиксирована коллизия. Так как компьютер сделал паузу после коллизии в 51,2 мкс, как это положено по стандарту (интервал отсрочки равен 512 битовых интервалов), а коммутатор — 50 мкс, то и в этом случае компьютеру не удалось передать свой кадр.

Коммутатор может пользоваться этим механизмом адаптивно, увеличивая степень своей агрессивности по мере необходимости.

Многие производители путем сочетания этих двух методов реализуют достаточно тонкие механизмы управления потоком кадров при перегрузках.

Простой отказ от поддержки алгоритма доступа к разделяемой среде без какой-либо модификации протокола ведет к повышению вероятности потерь кадров коммутаторами, так как при этом теряется контроль за потоками кадров, направляемых конечными узлами в сеть. При переходе на полнодуплексный режим узлу разрешается отправлять кадры в коммутатор всегда, когда это ему нужно, поэтому коммутаторы сети могут в этом режиме сталкиваться с перегрузками, не имея при этом никаких средств «притормаживания» потока кадров.

Причина перегрузок обычно кроется в ограниченной пропускной способности отдельного выходного порта, которая определяется параметрами протокола.

Поэтому, если входной трафик неравномерно распределяется между выходными портами, легко представить ситуацию, когда в какой-либо выходной порт коммутатора будет направляться трафик с суммарной средней интенсивностью большей, чем протокольный максимум. На рис. 17.3 показана как раз такая ситуация, когда в порт 3 коммутатора Ethernet направляется от портов 1, 2, 4 и 6 поток кадров размером в 64 байт с суммарной интенсивностью в 22 100 кадров в секунду. Вспомним, что максимальная скорость в кадрах в секунду для

сегмента Ethernet составляет 14 880. Естественно, что когда кадры поступают в буфер порта со скоростью 22 100 кадров в секунду, а уходят со скоростью 14 880 кадров в секунду, то внутренний буфер выходного порта начинает неуклонно заполняться необработанными кадрами.

Нетрудно подсчитать, что при размере буфера в 100 Кбайт в приведенном примере полное заполнение буфера произойдет через 0,22 с после начала работы в таком интенсивном режиме. Увеличение буфера до 1 Мбайт даст увеличение времени заполнения буфера до 2,2 с, что также неприемлемо. Проблему можно решить с помощью *средств контроля перегрузки*.

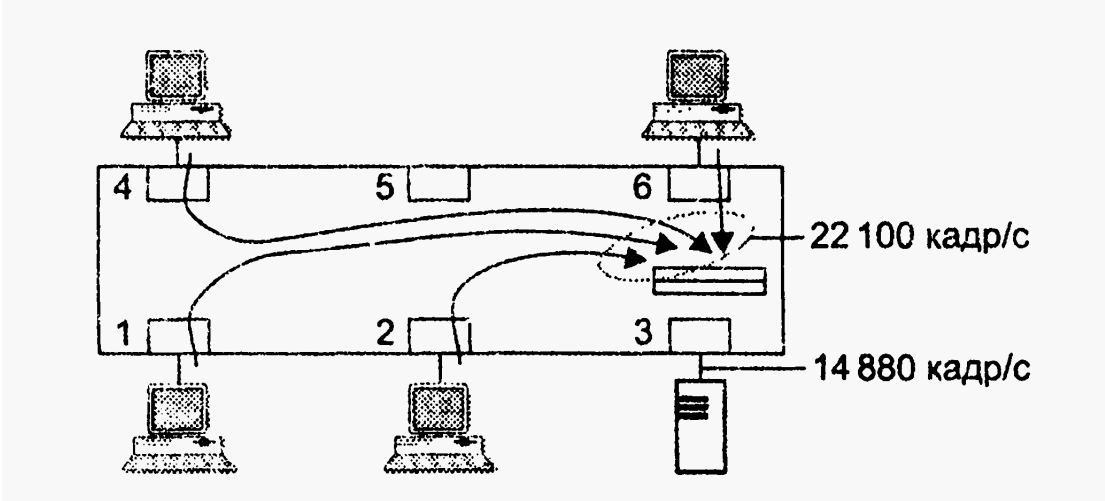


Рисунок 17.3. Переполнение буфера при несбалансированности трафика

Как мы знаем, существуют различные типы средств контроля перегрузки: управление очередями в коммутаторах, обратная связь, резервирование пропускной способности. На основе этих средств можно создать эффективную систему поддержки показателей QoS для трафика разных классов.

Механизм обратной связи был стандартизован для сетей Ethernet в марте 1997 года как спецификация IEEE 802.3x. Механизм обратной связи 802.3x используется только в дуплексном режиме работы портов коммутатора. Этот механизм очень важен для коммутаторов локальных сетей, так как он позволяет уменьшить потери кадров из-за переполнения буферов независимо от того, обеспечивает сеть дифференцированную поддержку показателей QoS для разных

типов трафика или же предоставляет базовый сервис по доставке с максимальными усилиями («по возможности»).

Спецификация 802.3x вводит новый подуровень в стеке протоколов Ethernet — **подуровень управления уровня MAC.** Он располагается над уровнем MAC и является необязательным (рис. 17.4).

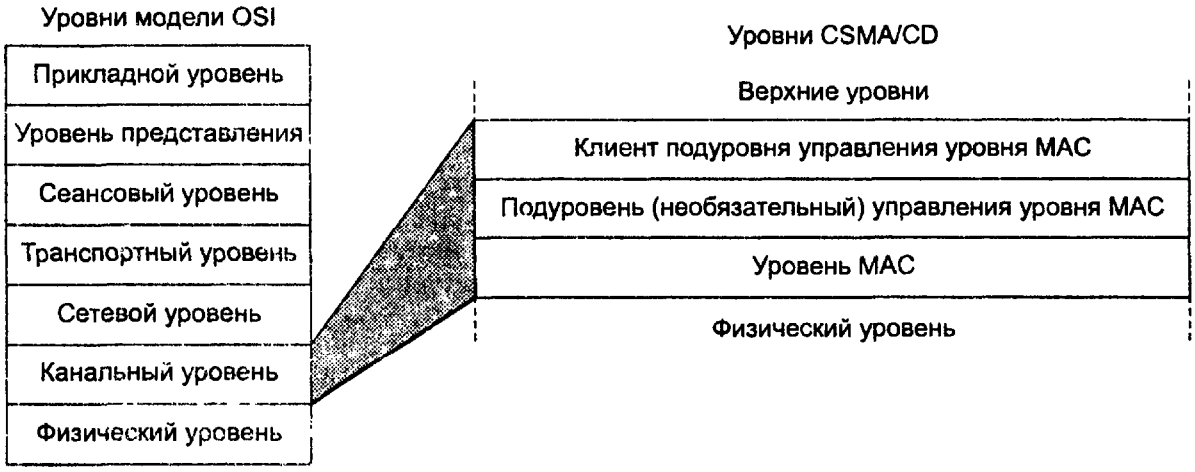


Рисунок 17.4. Подуровень управления уровня MAC

Кадры этого подуровня могут использоваться в различных целях, но пока в стандартах Ethernet для них определена только одна задача — приостановка передачи кадров другими узлами на определенное время.

Кадр подуровня управления отличается от кадров пользовательских данных тем, что в поле длины/типа всегда содержится шестнадцатеричное значение 88-08. Формат кадра подуровня управления рассчитан на универсальное применение, поэтому он достаточно сложен (рис. 17.5).

В качестве адреса назначения можно использовать зарезервированное для этой цели значение группового адреса 01-80-C2-00-00-01. Это удобно в том случае, когда соседний узел также является коммутатором (так как порты коммутатора не имеют уникальных MAC-адресов). Если сосед — конечный узел, можно также использовать уникальный MAC-адрес.

В поле кода операции подуровня управления указывается шестнадцатеричный код 00-01, поскольку, как уже было отмечено, пока

определена только одна операция подуровня управления, она называется *PAUSE* (пауза) и имеет шестнадцатеричный код 00-01.

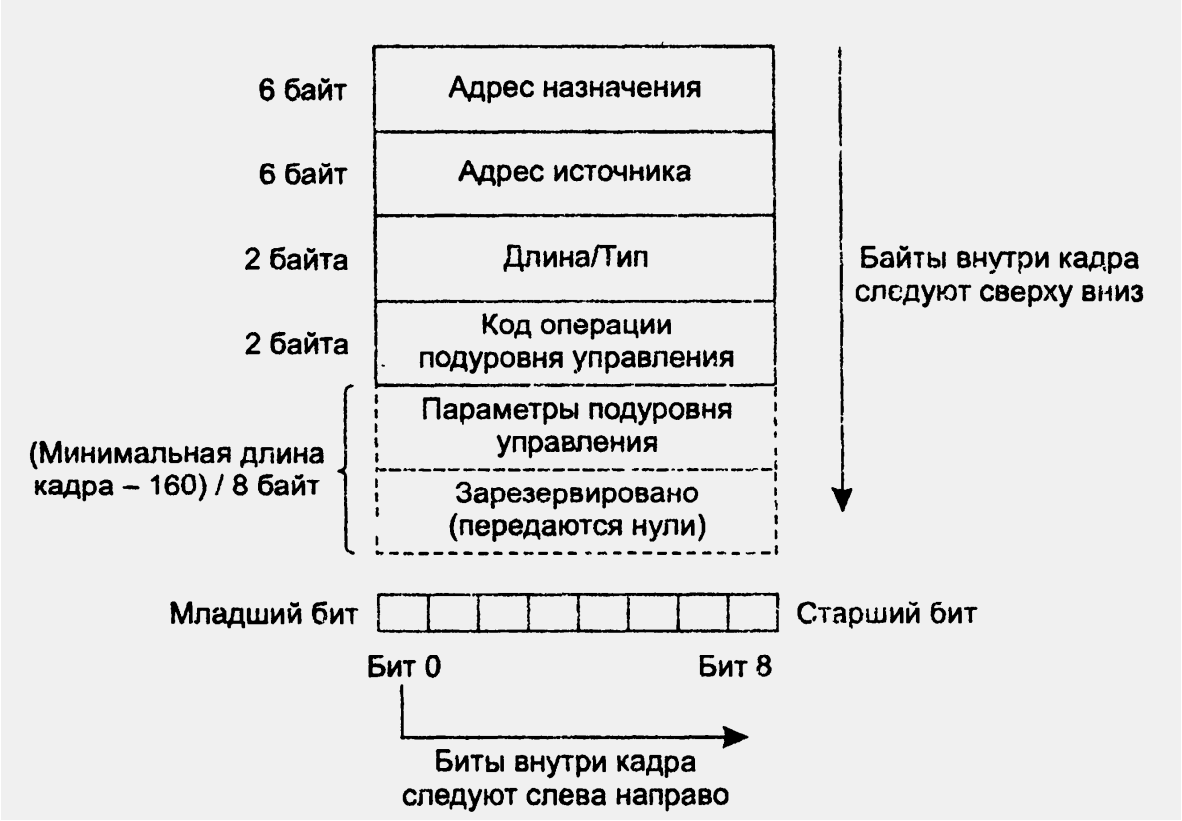


Рисунок 17.5. Формат кадра подуровни управления

В поле параметров подуровня управления указывается время, на которое узел, получивший такой код, должен прекратить передачу кадров узлу, отправившему кадр с операцией *PAUSE*. Время измеряется в 512 битовых интервалах конкретной реализации Ethernet, диапазон возможных вариантов приостановки равен 0-65535.

Как видно из описания, этот механизм обратной связи относится к типу 2. Специфика его состоит в том, что в нем предусмотрена только одна операция — приостановка на определенное время. Обычно же в механизмах этого типа используются две операции — приостановка и возобновление передачи кадров. Именно так этот механизм реализован в одном из наиболее старых протоколов сетей с коммутацией пакетов — протоколе сети X.25 под названием LAP-B.

18. АГРЕГИРОВАНИЕ ЛИНИЙ СВЯЗИ В ЛОКАЛЬНЫХ СЕТЯХ

18.1. Транки и логические каналы

Агрегирование линий связи (физических каналов) между двумя коммуникационными устройствами в один логический канал является еще одной формой использования избыточных альтернативных связей в локальных сетях.

При отказе одной из составляющих агрегированного логического канала, который часто называют **транком**, трафик распределяется между оставшимися линиями (рис. 18.1). На рисунке примером такой ситуации является транк 2, в котором один из физических каналов (центральный) отказал, так что все кадры передаются по оставшимся двум каналам. Этот пример демонстрирует повышение ***надежности*** при агрегировании.

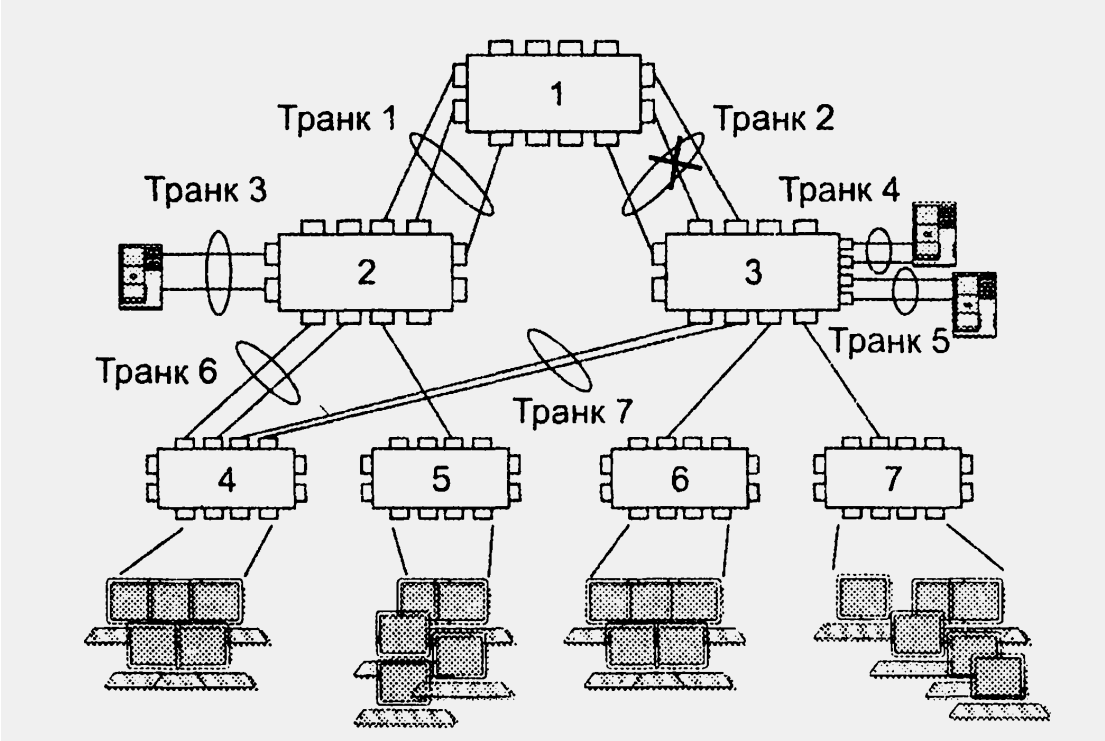


Рисунок 18.1. Агрегирование физических каналов

Покажем теперь, как агрегирование линий связи повышает ***производительность*** сети. Так, на рисунке коммутаторы 1 и 3 соединены тремя параллельными линиями связи, что в три раза повышает производительность

этого участка сети по сравнению со стандартным вариантом топологии дерева, которая не допускает таких параллельных связей. Повышение производительности связи между коммутаторами путем агрегирования линий связи в некоторых случаях является более эффективным, чем замена единственной линии связи более скоростной.

Агрегирование линий связи используется как для связей между портами коммутаторов локальной сети, так и для связей между компьютером и коммутатором. Чаще всего этот вариант выбирают для высокоскоростных и ответственных серверов. Для протокола IP или другого протокола сетевого уровня порты транка неразличимы, что соответствует концепции единого логического канала, лежащей в основе агрегирования.

Почти все методы агрегирования, применяемые в настоящее время, обладают существенным ограничением — в них учитываются только связи между двумя соседними коммутаторами сети и полностью игнорируется все, что происходит вне этого участка сети. Например, работа транка 1 никак не координируется с работой транка 2, и наличие обычной связи между коммутаторами 2 и 3, которая создает месте с транками 1 и 2 петлю, не учитывается. Поэтому технику агрегирования линий связи необходимо применять *одновременно* с алгоритмом покрывающего дерева — если администратор сети хочет использовать все топологические возможности объединения узлов сети. Для STA транк должен выглядеть как одна линия связи, тогда логика работы алгоритма останется в силе.

Существует большое количество фирменных реализаций механизма агрегирования линий связи. Наиболее популярные принадлежат, естественно, лидерам в секторе оборудования для локальных сетей. Это такие реализации, как Fast Ether-Channel и Gigabit Ether-Channel компании Cisco, MultiLink Trunking компании Nortel, Adaptive Load Balancing компании Intel и ряд других. Стандарт IEEE 802.3ad Link Aggregation обобщает эти подходы.

18.2. Борьба с «размножением» пакетов

Рассмотрим теперь подробней, в чем состоят особенности работы коммутатора в случае, когда его порты образуют транк. Во фрагменте сети, приведенном на рис. 18.2, два коммутатора — 1 и 2 — связаны четырьмя физическими каналами.

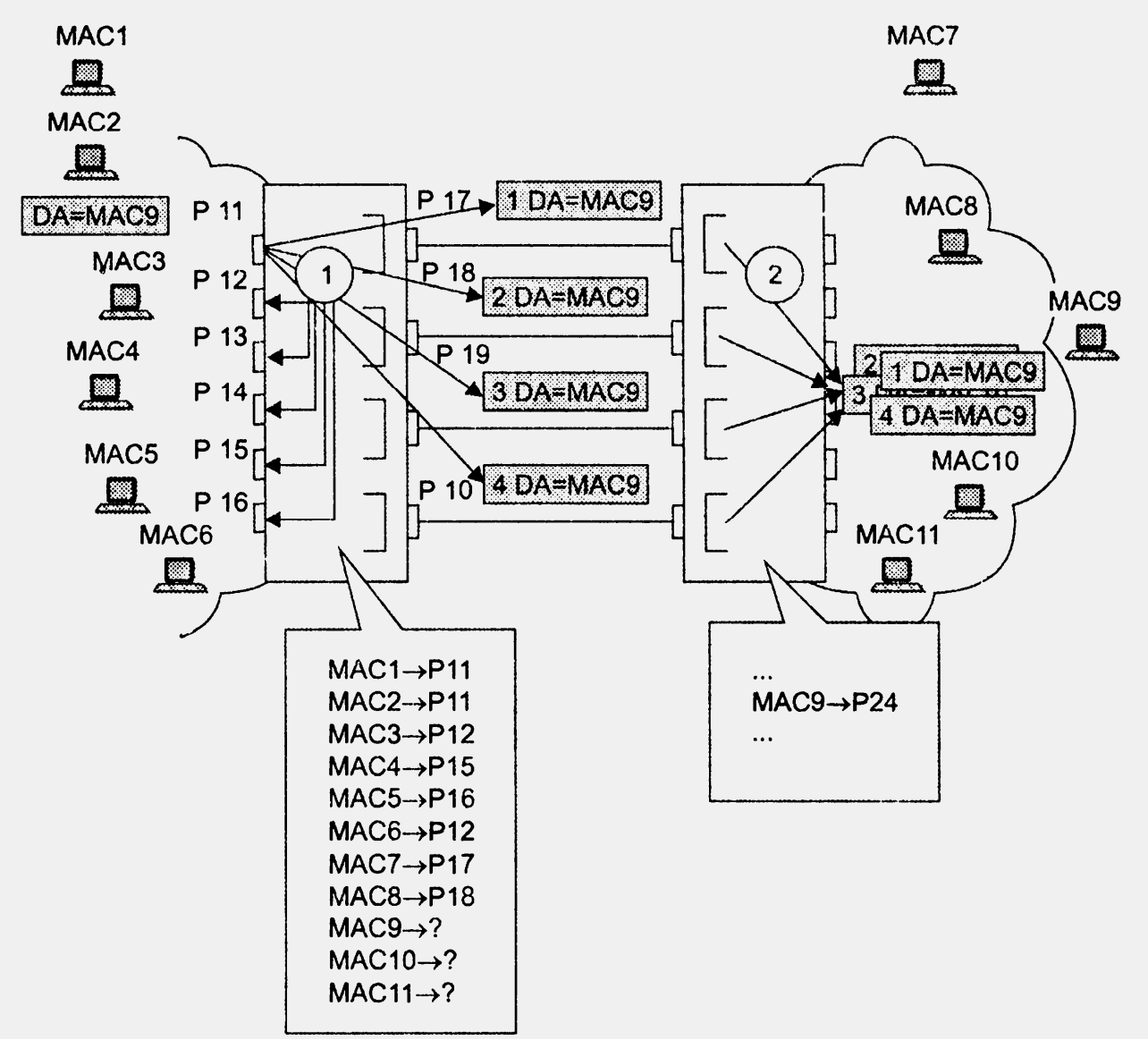


Рисунок 18.2. Размножение пакетов с неизученным адресом при наличии параллельных каналов между коммутаторами

Необходимо отметить, что транк может быть односторонним или двусторонним. Каждый коммутатор контролирует только отправку кадра, принимая решение, на какой из выходных портов его нужно передать. Поэтому

если оба коммутатора считают связывающие их каналы транком, то он будет двусторонним, в противном случае — односторонним.

Рисунок иллюстрирует поведение коммутатора 1 по отношению к параллельным каналам. В том случае, когда они не рассматриваются данным коммутатором как агрегированный канал, возникают проблемы с кадрами двух типов:

- кадрами с *еще не изученными* коммутатором уникальными адресами;
- кадрами, в которых указан *широковещательный* или *групповой адрес*.

Алгоритм прозрачного моста требует от коммутатора передавать кадр с неизученным (отсутствующим в таблице продвижения) адресом на все порты, кроме того, с которого кадр был принят. При наличии параллельных каналов такой кадр будет «размножен» в количестве, равном количеству каналов, — в приведенном примере коммутатор 2 примет четыре копии оригинального кадра.

При этом происходит также зацикливание кадров — они будут постоянно циркулировать между двумя коммутаторами, причем удалить их из сети окажется невозможно, так как в кадрах канального уровня отсутствует поле срока жизни, часто используемое в протоколах верхних уровней, например в IP и IPX.

В любом случае кадр с неизученным адресом повысит нагрузку на сеть за счет увеличения числа кадров, что чревато возникновением заторов, задержек и потерь данных. Помимо роста нагрузки дублирование кадров может привести также к неэффективной работе многих протоколов верхнего уровня.

Еще больше проблем создают кадры с широковещательным адресом — они всегда должны передаваться на все порты, кроме исходного, так что в любом случае «засорение» сети посторонним трафиком окажется значительным, и кадры будут зацикливаться.

С кадрами, у которых адрес назначения изучен, проблем у коммутаторов, связанных параллельными каналами, не возникает — коммутатор передает такой кадр на тот единственный порт, по которому этот кадр впервые пришел от источника.

Разработчики механизмов агрегирования учли проблемы, возникающие при обработке кадров с неизученными, широковещательными и групповыми адресами. Решение достаточно простое — все порты, связанные с параллельными каналами, считаются одним ***логическим портом***, который и фигурирует в таблице продвижения вместо нескольких ***физических портов***.

В примере, представленном на рис. 18.2, в таблице продвижения вместо портов P17, P18, P19 и P10 фигурирует логический порт AL11. С этим портом связаны адреса всех узлов, путь к которым лежит через коммутатор 2. При этом изучение нового адреса по кадру, поступившему от любого из физических портов, входящих в транк, приводит к появлению в таблице продвижения коммутатора новой записи с идентификатором логического порта. Поступающий в коммутатор кадр, адрес назначения которого изучен и связан с идентификатором логического порта, передается на один (и только один!) выходной физический порт, входящий в состав транка. Точно так же коммутатор поступает с неизученными, широковещательными и групповыми адресами — для передачи кадра используется только одна из связей. На порты коммутатора, не входящие в транк, это изменение в логике обработки кадров не распространяется. Так, коммутатор 1 всегда передает кадр с неизученным или широковещательным адресом на порты P11-P16. Благодаря такому решению кадры не дублируются и описанные проблемы не возникают.

18.3. Выбор порта

Остается открытым вопрос — какой из портов коммутатора нужно использовать для продвижения кадра через транк?

Можно предложить несколько вариантов ответов. Учитывая, что одной из целей агрегирования линий связи является повышение суммарной производительности участка сети между двумя коммутаторами (или коммутатором и сервером), следует распределять кадры по портам транка динамически, учитывая текущую загрузку каждого порта и направляя кадры в наименее загруженные (с меньшей длиной очереди) порты. **Динамический**

способ распределения кадров, учитывающий текущую загрузку портов и обеспечивающий баланс нагрузки между всеми связями транка, должен приводить, казалось бы, к максимальной пропускной способности транка.

Однако такое утверждение справедливо не всегда, так как в нем не учитывается поведение протоколов верхнего уровня. Существует ряд таких протоколов, производительность которых может существенно снизиться, если пакеты сеанса связи между двумя конечными узлами будут приходить не в том порядке, в котором они отправлялись узлом-источником. А такая ситуация может возникнуть, если два или более последовательных кадра одного сеанса будут передаваться через разные порты транка — по причине того, что очереди в буферах этих портов имеют разную длину. Следовательно, и задержка передачи кадра может быть разной, так что более поздний кадр обгонит более ранний.

Поэтому в большинстве реализаций механизмов агрегирования используются методы статического, а не динамического распределения кадров по портам. **Статический способ распределения кадров** подразумевает закрепление за определенным портом транка потока кадров определенного сеанса между двумя узлами, так что все кадры будут проходить через одну и ту же очередь и их упорядоченность не изменится.

Обычно при статическом распределении выбор порта для некоторого сеанса выполняется на основании определенных признаков, имеющихся в поступающих пакетах. Чаще всего такими признаками являются MAC-адреса источника или приемника, или оба вместе. В популярной реализации механизма Fast EtherChannel компании Cisco для коммутаторов семейства Catalyst 5000/6000 при выборе номера порта транка используется операция исключающего ИЛИ (XOR) над двумя последними битами MAC-адресов источника и приемника. Результат этой операции имеет четыре значения: 00, 01, 10 и 11, которые и являются условными номерами портов транка.

На рис. 18.3 приведен пример сети, в которой работает механизм Fast Ether-Channel.

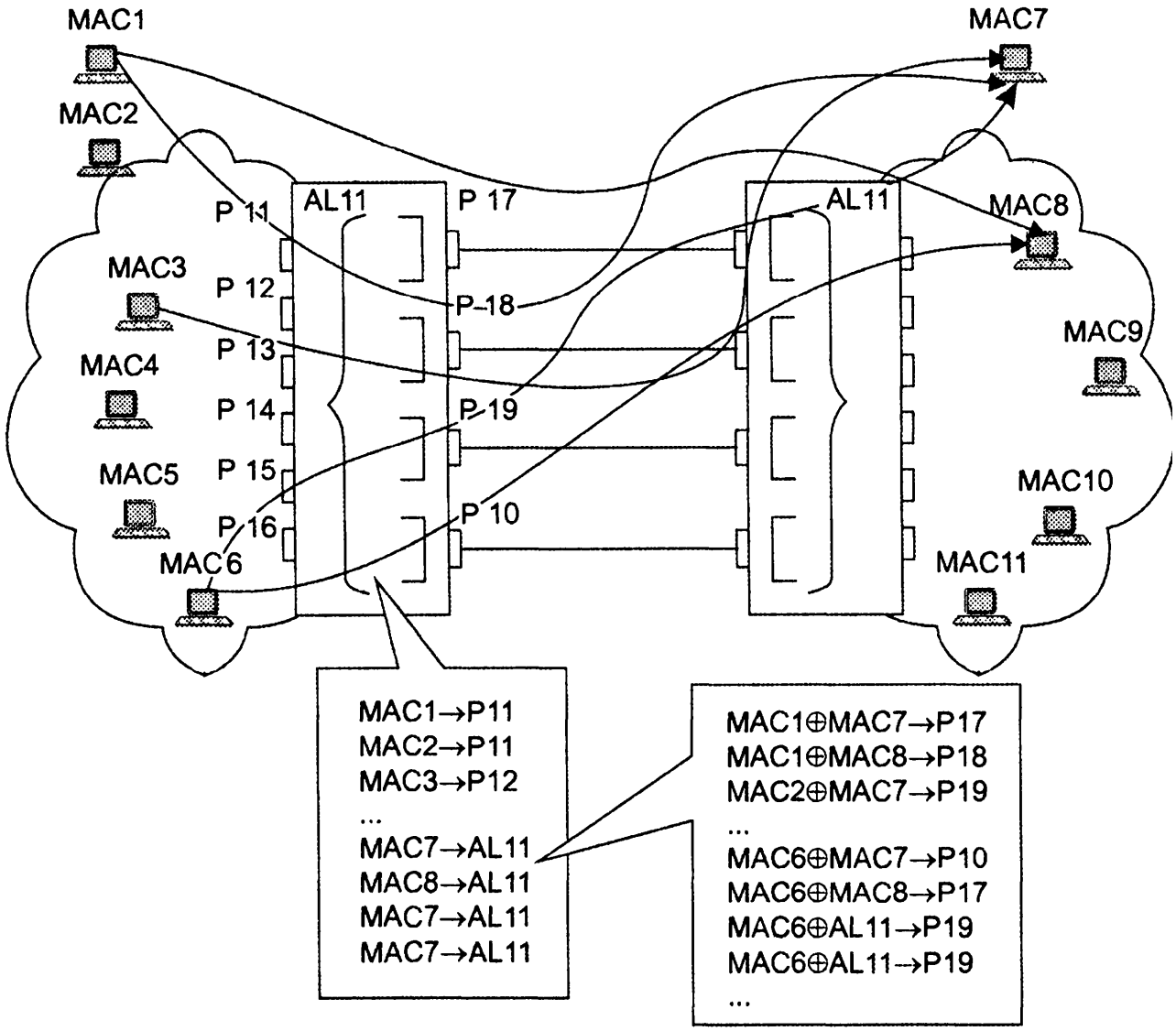


Рисунок 181.3. Пример сети с механизмом Fast Ether-Channel

Распределение потоков для сеансов между конечными узлами получается при этом достаточно случайным. Так как распределение не учитывает реальной нагрузки, которую создает каждый сеанс, общая пропускная способность транка может использоваться нерационально, особенно если интенсивности сеансов намного отличаются друг от друга. Кроме того, алгоритм распределения не гарантирует даже равномерного в количественном отношении распределения сеансов по портам. Случайный набор MAC-адресов в сети может привести к тому, что через один порт будут проходить несколько десятков сеансов, а через другой — только два-три. Выравнивание нагрузки портов можно при данном алгоритме

достигнуть только при большом количестве компьютеров и сеансов связи между ними.

Можно предложить и другие способы распределения сеансов по портам. Например, в соответствии с IP-адресами пакетов, которые инкапсулированы в кадры канального уровня, типами прикладных протоколов (почта по одному порту, веб-трафик по другому и т. д.). Полезным оказывается назначение порту сеансов с MAC-адресами, которые были изучены именно через этот порт — чтобы трафик сеанса проходил через один и тот же порт в обоих направлениях.

Стандартный способ создания агрегированных каналов, описанный в спецификации 802.3ad, предполагает возможность создания логического порта путем объединения нескольких физических портов, принадлежащих разным коммутаторам. Для того чтобы коммутаторы могли автоматически обеспечиваться информацией о принадлежности какого-либо физического порта определенному логическому порту, в спецификации предложен служебный **протокол управления агрегированием линий связи** (Link Control Aggregation Protocol, LCAP).

При отказе какого-либо канала транка все пакеты сеансов, назначенные для соответствующего порта, будут направляться на один из оставшихся портов. Обычно восстановление связности при таком отказе занимает от единиц до десятков миллисекунд. Это объясняется тем, что во многих реализациях транка после отказа физического канала все MAC-адреса, которые были с ним связаны, принудительно помечаются как неизученные. Затем коммутатор повторяет процедуру изучения этих адресов. После этого процедура назначения сеанса портам выполняется заново, естественно, учитываются только работающие порты. Так как тайм-ауты в сеансах протоколов локальных сетей обычно небольшие, коротким оказывается и время восстановления соединения.

19. ВИРТУАЛЬНЫЕ ЛОКАЛЬНЫЕ СЕТИ

Важным свойством коммутатора локальной сети является способность контролировать передачу кадров между сегментами сети. По различным причинам (соблюдение прав доступа, политика безопасности и т. д.) некоторые кадры не следует передавать по адресу назначения.

Такого типа ограничения можно реализовать с помощью *пользовательских фильтров*. Однако пользовательский фильтр может запретить коммутатору передачу кадров только по конкретным адресам, а широковещательный трафик он *обязан* передать всем сегментам сети. Так требует алгоритм его работы. Поэтому сети, созданные на основе коммутаторов, иногда называют *плоскими* — из-за отсутствия барьеров на пути широковещательного трафика. Технология виртуальных локальных сетей позволяет преодолеть указанное ограничение.

Виртуальной локальной сетью (VLAN) – называется группа узлов сети, трафик которой, в том числе и широковещательный на канальном уровне полностью изолирован от трафика других узлов сети.

Это означает, что передача кадров между разными виртуальными сетями на основании адреса канального уровня невозможна независимо от типа адреса — уникального, группового или широковещательного. В то же время внутри виртуальной сети кадры передаются по технологии коммутации, то есть только на тот порт, который связан с адресом назначения кадра.

Виртуальные локальные сети могут *перекрываться*, если один или несколько компьютеров входят в состав более чем одной виртуальной сети. На рис. 19.1 сервер электронной почты входит в состав виртуальных сетей 3 и 4. Это означает, что его кадры передаются коммутаторами всем компьютерам, входящим в эти сети. Если же какой-то компьютер входит в состав только виртуальной сети 3, то его кадры до сети 4 доходить не будут, но он может взаимодействовать с компьютерами сети 4 через общий почтовый сервер. Такая схема не полностью защищает виртуальные сети друг от друга — так,

широковещательный шторм, возникший на сервере электронной почты, затопит и сеть 3, и сеть 4.

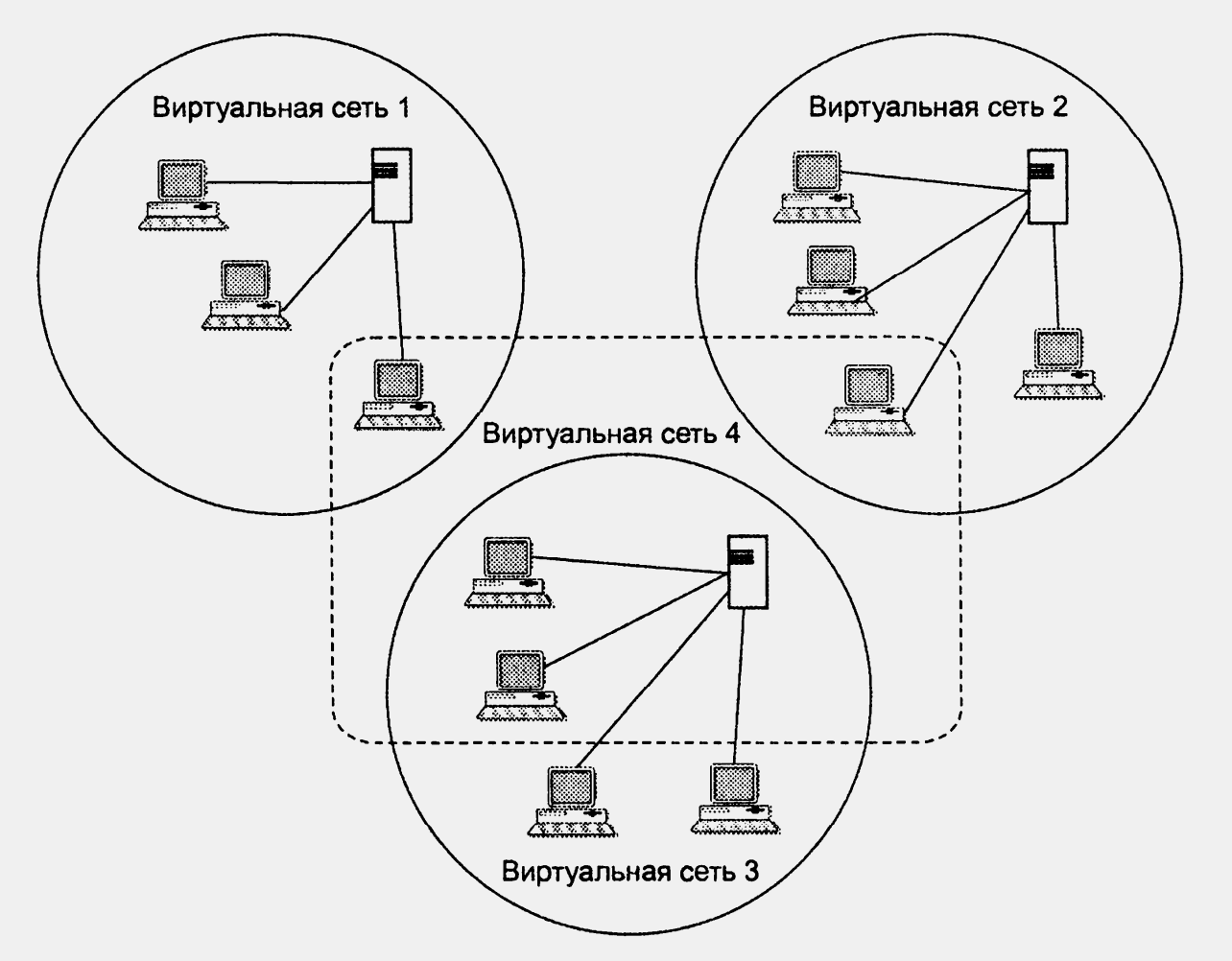


Рисунок 19.1. Виртуальные локальные сети

Говорят, что виртуальная сеть образует ***домен широковещательного трафика***, по аналогии с доменом коллизий, который образуется повторителями сетей Ethernet.

19.1. Назначение виртуальных сетей

Основное назначение технологии VLAN состоит в облегчении процесса создания изолированных сетей, которые затем обычно связываются между собой с помощью маршрутизаторов. Такое построение сети создает мощные барьеры на пути нежелательного трафика из одной сети в другую. Сегодня считается очевидным, что любая крупная сеть должна включать маршрутизаторы, иначе потоки ошибочных кадров, например широковещательных, будут периодически

«затапливать» всю сеть через прозрачные для них коммутаторы, приводя ее в неработоспособное состояние.

Достоинством технологии виртуальных сетей является то; что она позволяет создавать полностью изолированные сегменты сети, путем логического конфигурирования коммутаторов не прибегая к изменению физической структуры.

До появления технологии VLAN для создания отдельной сети использовались либо физически изолированные сегменты коаксиального кабеля, либо несвязанные между собой сегменты, построенные на повторителях и мостах. Затем эти сети связывались маршрутизаторами в единую составную сеть (рис. 19.2).

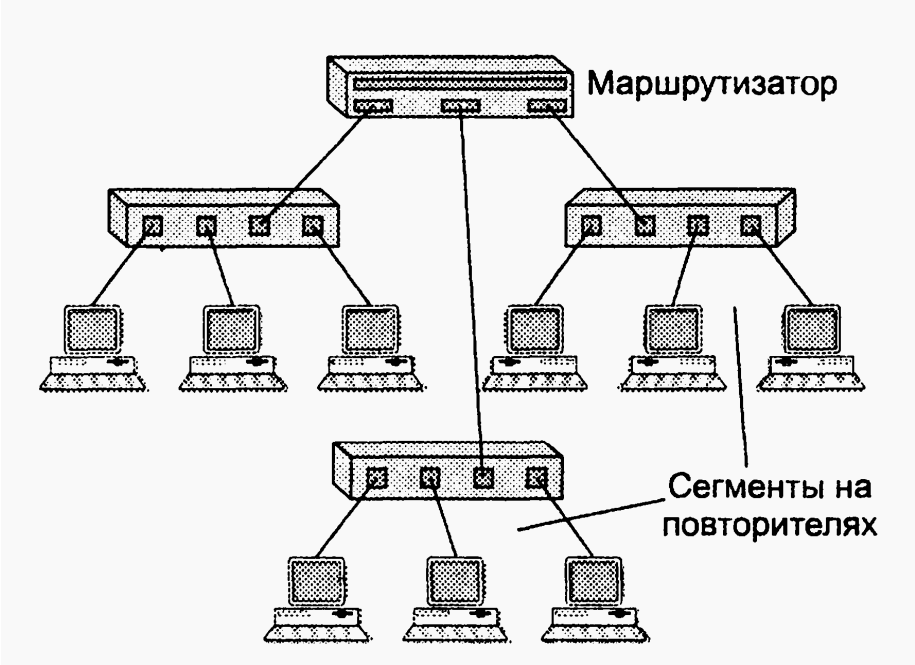


Рисунок 19.2.- Составная сеть, состоящая из сетей, построенных на основе повторителей

Изменение состава сегментов (переход пользователя в другую сеть, дробление крупных сегментов) при таком подходе подразумевает физическую перекоммутацию разъемов на передних панелях повторителей или в кроссовых панелях, что не очень удобно в больших сетях — много физической работы, к тому же высока вероятность ошибки.

Для связи виртуальных сетей в общую сеть требуется привлечение сетевого уровня. Он может быть реализован в отдельном маршрутизаторе, а может работать и в составе программного обеспечения коммутатора, который тогда становится комбинированным устройством — так называемым **коммутатором 3-го уровня.**

Технология виртуальных сетей долгое время не стандартизировалась, хотя и была реализована в очень широком спектре моделей коммутаторов разных производителей. Положение изменилось после принятия в 1998 году стандарта IEEE 802.1Q, который определяет базовые правила построения виртуальных локальных сетей, не зависящие от протокола канального уровня, поддерживаемого коммутатором.

19.2. Создание виртуальных сетей на базе одного коммутатора

При создании виртуальных сетей на основе одного коммутатора обычно используется механизм ***группирования портов*** коммутатора (рис. 19.3). При этом каждый порт приписывается той или иной виртуальной сети. Кадр, пришедший от порта, принадлежащего, например, виртуальной сети 1, никогда не будет передан порту, который не принадлежит этой виртуальной сети. Порт можно приписать нескольким виртуальным сетям, хотя на практике так делают редко — пропадает эффект полной изоляции сетей.

Если к порту коммутатора подключен сегмент, построенный на основе повторителя, то узлы такого сегмента не имеет смысла включать в разные виртуальные сети — все равно трафик этих узлов будет общим.

Создание виртуальных сетей путем группирования портов не требует от администратора большого объема ручной работы — достаточно каждый порт приписать к одной из нескольких заранее поименованных виртуальных сетей. Обычно такая операция выполняется с помощью специальной программы, прилагаемой к коммутатору.

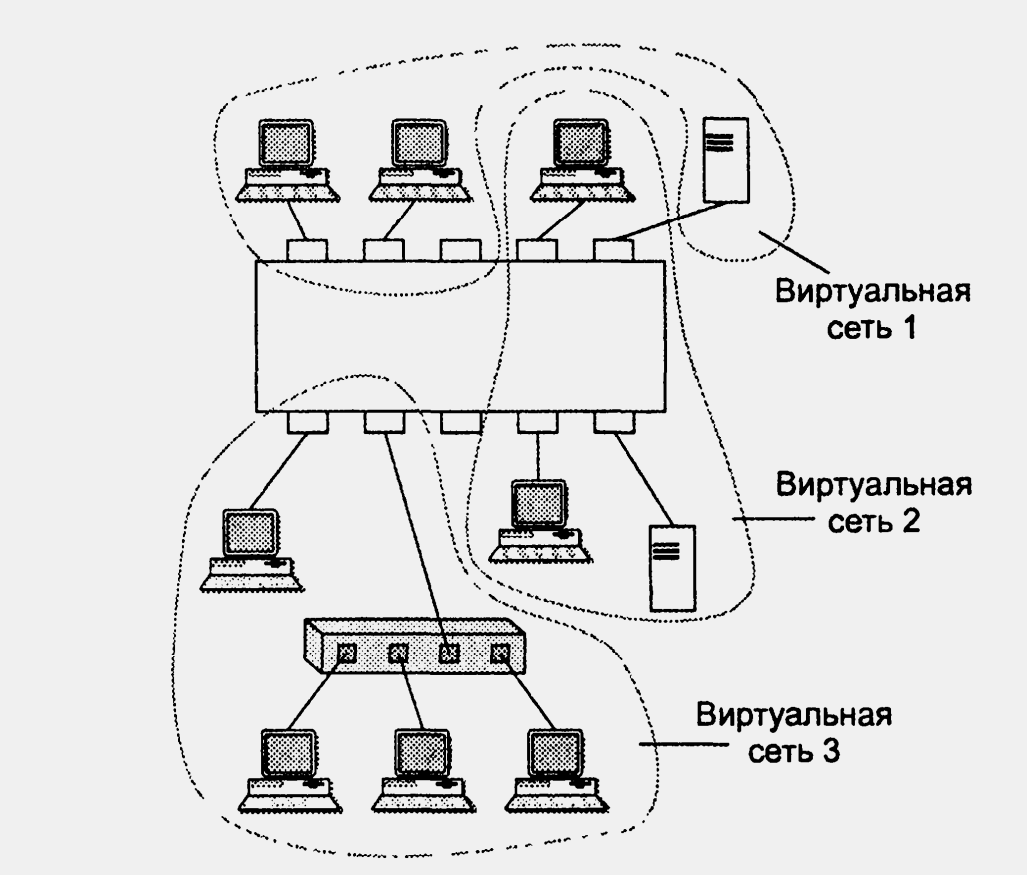


Рисунок 19.3. Виртуальные сети, построенные на основе одного коммутатора

Второй способ образования виртуальных сетей основан на ***группировании MAC-адресов***. Каждый MAC-адрес, который изучен коммутатором, приписывается той или иной виртуальной сети. При существовании в сети множества узлов этот способ требует от администратора выполнения большого количества ручных операций. Однако при построении виртуальных сетей на основе нескольких коммутаторов он оказывается более гибким, чем группирование портов.

19.3. Создание виртуальных сетей на базе нескольких коммутаторов

Рисунок 19.4 иллюстрирует проблему, возникающую при создании виртуальных сетей на основе нескольких коммутаторов, поддерживающих технику ***группирования портов***.

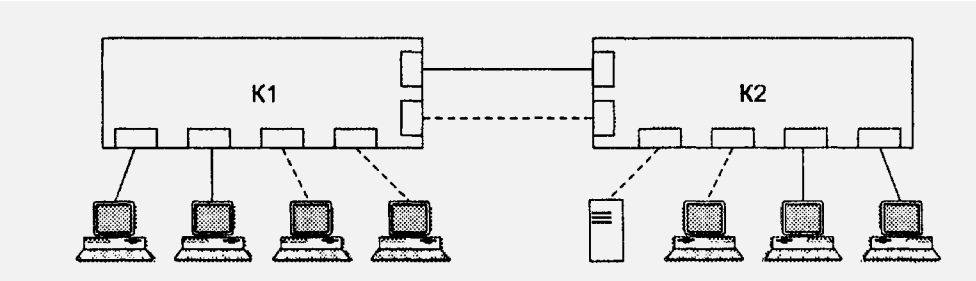


Рисунок 19.5. Построение виртуальных локальных сетей на нескольких коммутаторов с группированием портов

Если узлы какой-либо виртуальной сети подключены к разным коммутаторам, то для подключения каждой такой сети на коммутаторах должна быть выделена специальная пара портов. В противном случае, если коммутаторы будут связаны только одной парой портов, информация о принадлежности кадра той или иной виртуальной сети при передаче из коммутатора в коммутатор будет утеряна. Таким образом, коммутаторы с группированием портов требуют для своего соединения столько портов, сколько виртуальных сетей они поддерживают. Порты и кабели используются при таком способе очень расточительно. Кроме того, при соединении виртуальных сетей через маршрутизатор для каждой виртуальной сети выделяется отдельный кабель и отдельный порт маршрутизатора, что также приводит к большим накладным расходам.

Группирование MAC-адресов в виртуальную сеть на каждом коммутаторе избавляет от необходимости связывать их по нескольким портам, поскольку в этом случае MAC-адрес является меткой виртуальной сети. Однако этот способ требует выполнения большого количества ручных операций по маркировке MAC-адресов на каждом коммутаторе сети.

Описанные два подхода основаны только на добавлении дополнительной информации к адресным таблицам коммутатора, и в них отсутствует возможность встраивания в передаваемый кадр информации о принадлежности кадра виртуальной сети. В остальных подходах используются имеющиеся или дополнительные поля кадра для сохранения информации о принадлежности кадра

той или иной виртуальной локальной сети при его перемещениях между коммутаторами сети. При этом нет необходимости запоминать в каждом коммутаторе принадлежность всех MAC-адресов составной сети виртуальным сетям.

Дополнительное поле с пометкой о номере виртуальной сети используется только тогда, когда кадр передается от коммутатора к коммутатору, а при передаче кадра конечному узлу оно обычно удаляется. При этом модифицируется протокол взаимодействия «коммутатор-коммутатор», а программное и аппаратное обеспечение конечных узлов остается неизменным. До принятия стандарта IEEE 802.1Q существовало много фирменных протоколов этого типа, но все они имели один недостаток — оборудование различных производителей оказывалось несовместимым при образовании VLAN.

В стандарте **IEEE 802.1Q** для хранения номера виртуальной сети предусмотрен дополнительный заголовок в 2 байта, который этот протокол делит с протоколом 802.1p. Помимо 3 бит для хранения приоритета кадра, описанных стандартом 802.1p, в этом заголовке 12 бит используются для хранения номера виртуальной сети, к которой принадлежит кадр. Эта дополнительная информация, которая называется **тегом виртуальной сети,** позволяет коммутаторам разных производителей создавать до 4096 общих виртуальных сетей. Кадр с такой информацией называют **«помеченным»** или **тегированным**. Длина помеченного кадра Ethernet увеличивается на 4 байта, так как помимо 2 байт собственно тега добавляются еще 2 байта. Структура помеченного кадра Ethernet показана на рис. 19.6. При добавлении заголовка 802.1p/Q поле данных уменьшается на 2 байта.

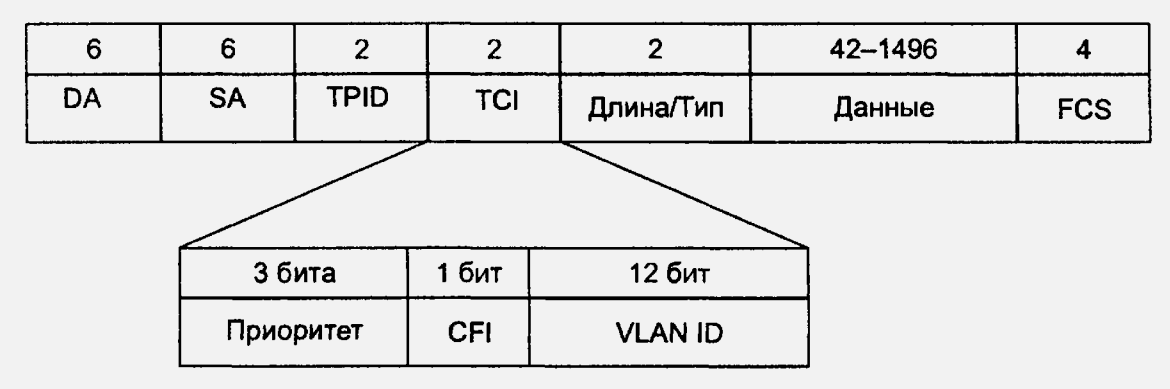


Рисунок 19.6. Структура помеченного кадра Ethernet

Введение стандарта 802.1Q позволило производителям оборудования преодолеть различия в фирменных реализациях VLAN и добиться совместимости при построении виртуальных локальных сетей. Поддерживают технику VLAN как производители коммутаторов, так и сетевых адаптеров. В последнем случае сетевой адаптер может генерировать и принимать помеченные кадры Ethernet, содержащие поле тега виртуальной сети. Если сетевой адаптер генерирует помеченные кадры, то тем самым он определяет их принадлежность к той или иной виртуальной локальной сети, поэтому коммутатор должен обрабатывать их соответствующим образом, то есть передавать или не передавать на выходной порт в зависимости от принадлежности порта. Драйвер сетевого адаптера может получить номер своей (или своих) виртуальной локальной сети путем ручного конфигурирования либо от некоторого приложения, работающего на данном узле или на одном из серверов сети.

Ограничения мостов и коммутаторов

Применение коммутаторов позволяет преодолеть ограничения, свойственные сетям с разделяемой средой. Коммутируемые локальные сети могут покрывать значительные территории, плавно переходя в сети мегаполисов; они могут состоять из сегментов различной пропускной способности, образуя сети с очень высокой производительностью; они могут использовать альтернативные маршруты для повышения надежности и производительности. Однако построение сложных сетей только на основе повторителей, мостов и коммутаторов, то есть без применения устройств сетевого уровня, имеет существенные ограничения.

1. Серьезные ограничения по-прежнему накладываются на топологию коммутируемой локальной сети. Требование *отсутствия петель* преодолевается с помощью техники STA и агрегирования каналов только частично. Действительно, STA не позволяет использовать все альтернативные маршруты для передачи пользовательского трафика, а агрегирование каналов разрешает так делать только на участке сети между двумя соседними коммутаторами. Подобные

ограничения не позволяют применять многие эффективные топологии, которые могли бы использоваться для передачи трафика.

2. Логические сегменты сети, расположенные между коммутаторами, *слабо изолированы* друг от друга, а именно не защищены от так называемых широковещательных штормов. Использование же механизма виртуальных сетей, реализованного во многих коммутаторах, хотя и позволяет достаточно гибко создавать изолированные по трафику группы станций, но при этом изолирует их полностью, то есть так, что узлы одной виртуальной сети не могут взаимодействовать с узлами другой виртуальной сети.

3. В сетях, построенных на основе мостов и коммутаторов, достаточно *сложно решается задача фильтрации трафика* на основе данных, содержащихся в пакете. В таких сетях фильтрации выполняется только с помощью пользовательских фильтров, для создания которых администратору приходится иметь дело с двоичным представлением содержимого пакетов.

4. Реализация транспортной подсистемы только средствами физического и канального уровней приводит к *недостаточно гибкой*, *одноуровневой системе адресации:* в качестве адреса назначения используется MAC-адрес, жестко связанный с сетевым адаптером.

5. У коммутаторов *ограничены возможности по трансляции протоколов* при создании гетерогенной сети. Они не могут транслировать WAN-протоколы в LAN-протоколы из-за различий в системе адресации этих сетей, а также различных значений максимального размера поля данных.

Наличие серьезных ограничений у протоколов канального уровня показывает, что построение на основе средств этого уровня больших неоднородных сетей является весьма проблематичным. Естественное решение в этих случаях — привлечение средств более высокого, сетевого уровня.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. В.Г.Олифер, Н.А.Олифер. Компьютерные сети. Учебник. СПб:. Питер, 2006.– 672 с.
2. В. Столлингс. Современные компьютерные сети. СПб:. Питер, 2003. – 783 с.
3. Майкл Дж. Мартин. Введение в сетевые технологии. Практическое руководство по организации сетей. М., 2002.- 546 с.
4. М. Спортак, Ф. Поппас. Компьютерные сети и сетевые технологии. Киев, 2002.- 736 с.
5. В. Столлингс. Компьютерные системы передачи данных. Москва, 2002.-928 с.
6. Майкл Палмер, Роберт Брюс Синклер. Проектирование и внедрение компьютерных сетей. Учебный курс. СПб:. Питер, 2004.- 578 с.
7. Шиндер Дебра Литлджон. Основы компьютерных сетей. М., 2003.- 623 с.
8. Шин Одом, Хенсон Ноттингем. Коммутаторы CISCO. М., Кудиц-Образ, 2003.- 528 с.
9. И. Руденко. Маршрутизаторы CISCO для IP-сетей. М., Кудиц-Образ, 2003.- 656 с.
10. Кларк Кеннеди, Гамильтон Кевин. Принципы коммутации в локальных сетях CISCO. М., Кудиц-Образ, 2003.- 976 с.
11. Семенов А.Б. Проектирование и расчет структурированных кабельных систем и их компонентов. М, Ай Ти, 2003.- 416 с.

Учебное издание

Конспект лекций по курсу

«Компьютерные сети»

Для студентов, обучающихся по направлению

6.050903 “Телекомуникации”

(для дневной и заочной форм обучения)

Составители: Федюн Роман Валериевич, к.т.н., доцент

Попов Владислав Александрович, к.т.н., доцент

Рецензент Светличная Виктория Антоновна, к.т.н., доцент

Червинский Владимир Владимирович, к.т.н., доцент

Отв. за выпуск Воропаева Виктория Яковлевна, к.т.н., доцент, зав .каф.