

Министерство образования Российской Федерации

**Федеральное государственное образовательное учреждение среднего
профессионального образования
«Уральский радиотехнический техникум им. А.С. Попова»**

Курс лекций дисциплины

КОМПЬЮТЕРНЫЕ СЕТИ И ТЕЛЕКОММУНИКАЦИИ

(версия 3)

Разработал В.Г. Аверин

**Екатеринбург
2009 г.**

СОДЕРЖАНИЕ

Глава 1.	Эволюция компьютерных сетей	2
	Тема 1.1. Вычислительная и телекоммуникационные технологии	2
	Тема 1.2. Системы пакетной обработки	3
	Тема 1.3. Многотерминальные системы – прообраз сети	4
	Тема 1.4. Первые глобальные сети	5
	Тема 1.5. Первые локальные сети	7
	Тема 1.6. Сближение локальных и глобальных сетей	11
	Тема 1.7. Конвергенция компьютерных и телекоммуникационных сетей .	12
	Выводы	14
	Вопросы и задания	15
Глава 2.	Общие принципы построения сетей	16
	Тема 2.1. Связь компьютера с периферийными устройствами	16
	Тема 2.2. Простейший случай взаимодействия двух компьютеров	18
	Тема 2.3. Сетевые службы и приложения	21
	Тема 2.4. Физическая передача данных по линиям связи	23
	Тема 2.5. Характеристики физических каналов	26
	Тема 2.6. Топология физических связей	28
	Тема 2.7. Адресация узлов сети	31
	Тема 2.8. Коммутация	34
	Тема 2.9. Определение информационных потоков	35
	Тема 2.10. Маршрутизация	35
	Тема 2.11. Продвижение данных	36
	Тема 2.12. Мультиплексирование и демultipлексирование	36
	Тема 2.13. Разделяемая среда передачи данных	38
	Тема 2.14. Типы коммутации	41
	Выводы	41
	Вопросы и задания	42
Глава 3.	Архитектура и стандартизация сетей.....	44
	Тема 3.1. Декомпозиция задачи сетевого взаимодействия	44
	Тема 3.2. Многоуровневый подход	45
	Тема 3.3. Протокол и стек протоколов	45
	Тема 3.4. Общая характеристика модели OSI	46
	Тема 3.5. Физический уровень	49
	Тема 3.6. Канальный уровень	49
	Тема 3.7. Сетевой уровень	51
	Тема 3.8. Транспортный уровень	54
	Тема 3.9. Сеансовый уровень	55
	Тема 3.10. Уровень представления	55
	Тема 3.11. Прикладной уровень	55
	Тема 3.12. Модель OSI и сети с коммутацией каналов	55
	Тема 3.13. Сетевая технология Ethernet.....	56
	Тема 3.14. Стек TCP/IP	58
	Тема 3.15. Соответствие популярных стеков протоколов модели OSI.....	62
	Тема 3.16. Информационные и транспортные услуги	63
	Тема 3.17. Распределение протоколов по элементам сети	64
	Тема 3.18. Вспомогательные протоколы транспортной системы	66
	Выводы	67
	Вопросы и задания	68

Глава 4.	Коммутация пакетов и каналов	70
	Тема 4.1. Коммутация каналов	70
	Тема 4.2. Установление соединения	71
	Тема 4.3. Отказ в установлении соединения	71
	Тема 4.4. Гарантированная пропускная способность	71
	Тема 4.5. Мультиплексирование	72
	Тема 4.6. Неэффективность передачи пульсирующего трафика	73
	Тема 4.7. Коммутация пакетов	74
	Тема 4.8. Буферы и очереди	75
	Тема 4.9. Методы продвижения пакетов	77
	Тема 4.10. Дейтаграммная передача	78
	Тема 4.11. Логическое соединение	79
	Тема 4.12. Виртуальный канал	81
	Тема 4.13. Сравнение сетей с коммутацией пакетов и каналов	82
	Тема 4.14. Разделение среды	82
	Тема 4.15. Причины структуризации локальных сетей	84
	Тема 4.16. Физическая структуризация локальной сети	84
	Тема 4.17. Логическая структуризация сети на разделяемой среде	87
	Выводы	91
	Вопросы и задания	92

Литература.

1. В. Олифер, Н. Олифер. Компьютерные сети. 3-е издание. Учебник. С-Петербург, Питер, 2006 г.

АННОТАЦИЯ

Компьютерные сети, интенсивно развивающаяся часть информационных сетей, в своем развитии прошли и проходят следующие этапы:

1. Первый этап – с момента создания и ориентировочно до середины 90-х годов прошлого столетия локальные и глобальные сети существовали отдельно.
2. Второй этап – с середины 90-х годов, постепенно, различия между локальными и глобальными типами сетевых технологий стали сглаживаться, и к середине первого десятилетия 2000-х годов тесная интеграция локальных и глобальных сетей привела к значительному взаимопроникновению соответствующих технологий.
3. Третий этап – с середины первого десятилетия 2000-х годов усиливается тенденция сближения компьютерных и телекоммуникационных (телефонные, радио, телевизионные) сетей разных видов.

Естественно, вышеназванные причины значительно влияют на подход к изложению материала по изучению информационных (компьютерных и телекоммуникационных) сетей. Изменяется терминология, появляются новые технологии, требуется дополнительно пояснять материал из других научно-технических отраслей.

Все это побудило автора разработать данную версию курса лекций «Компьютерные сети и телекоммуникации», значительно отличающуюся от предыдущих версий.

Основой лекционного материала послужил учебник авторов В.Г. Олифер, Н.А. Олифер «Компьютерные сети» (принципы, технологии, протоколы, 3-е издание, 2006 год издания, издательство Питер).

Курс лекций охватывает больший объем материала, чем требуется в Государственных стандартах СПО, так как, по мнению автора, требования стандартов по изучению информационных сетей были разработаны в рамках второго этапа развития компьютерных сетей и потому в какой-то степени устарели.

В комплект курса лекций входит:

1. Конспект лекций (в электронном и бумажном варианте), разбитый на 4-е главы (блока). В конце каждой главы приведены вопросы и задания для проверки уровня усвоения материала.
2. Слайды с рисунками для демонстрации через проектор при чтении лекционного материала.
3. Вопросы электронного тестирования по каждой главе. База данных и тесты разработаны в системе «ПоЗнание» версии 3.0. Система тестового контроля позволяет проверять знания студентов по окончании изучения каждой главы, на экзаменах по окончании курса, на междисциплинарных экзаменах по окончании обучения.

Блочная структура построения лекционного материала позволяет преподавать по дисциплинам, касающимся информационных сетей, на следующих специальностях:

1. 210306 «Радиоаппаратостроение».
2. 230101 «Вычислительные машины, комплексы, системы и сети».
3. 230103 «Автоматизированные системы обработки информации и управления».
4. 230105 «Программное обеспечение вычислительной техники и автоматизированных систем»
5. 230106 «Техническое обслуживание средств вычислительной техники и компьютерных сетей».
6. 080802 «Прикладная информатика».
7. 090108 «Информационная безопасность».

Главы 1 и 2 обязательны для изучения на всех специальностях. Главы 3 и 4 используются в зависимости от специальности в разной комплектации.

Курс лекций прошел апробацию в 2008-2009 учебном году при проведении занятий по специальностям 230101, 230103, 230105, 230106, 080802.

ГЛАВА 1 Эволюция компьютерных сетей

Компьютерные сети появились сравнительно недавно, в конце 60-х годов. Естественно, что компьютерные сети унаследовали много полезных свойств от других, более старых и распространенных телекоммуникационных сетей, а именно телефонных.

Вычислительная и телекоммуникационная технологии

Компьютерные сети, называемые также **сетями передачи данных**, являются логическим результатом эволюции двух важнейших научно технических отраслей современной цивилизации — компьютерных и телекоммуникационных технологий. (рис. 1.1)

С одной стороны, сети представляют собой частный случай распределенных вычислительных систем, в которых группа компьютеров согласованно выполняет набор взаимосвязанных задач, обмениваясь данными в автоматическом режиме.

Распределенными вычислительными системами называются системы, имеющие несколько центров обработки информации.

Поскольку основным признаком **распределенной вычислительной системы** является наличие нескольких центров обработки данных, то наряду с компьютерными сетями к распределенным системам относятся также **мультипроцессорные компьютеры** и **многомашинные вычислительные комплексы**.

В **мультипроцессорных компьютерах** имеется несколько процессоров, каждый из которых может относительно независимо от остальных выполнять свою программу.

Многомашинная система – это вычислительный комплекс, включающий в себя несколько компьютеров (каждый из которых работает под управлением собственной операционной системы), а также программные и аппаратные средства связи компьютеров, которые обеспечивают работу всех компьютеров комплекса как единого целого.

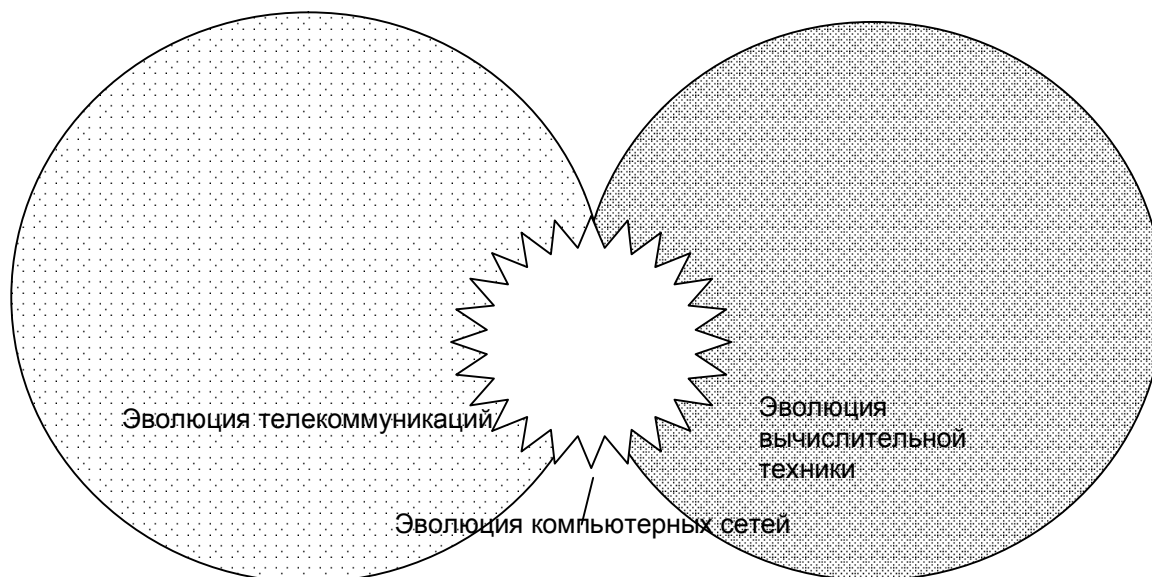


Рис. 1.1. Эволюция компьютерных сетей на стыке вычислительной техники и телекоммуникационных технологий

В **вычислительных сетях** автономность обрабатывающих блоков проявляется в наибольшей степени – основными элементами сети являются стандартные компьютеры, не имеющие ни общих блоков памяти, ни общих периферийных устройств. Связь между

компьютерами осуществляется с помощью специальных периферийных устройств – сетевых адаптеров, соединенных относительно протяженными каналами связи.

С другой стороны, компьютерные сети могут рассматриваться как средство передачи информации на большие расстояния, для чего в них применяются методы кодирования и мультиплексирования данных, получившие развитие в различных телекоммуникационных системах – телефония, радио- и теле- вещание.

Системы пакетной обработки

Обратимся сначала к компьютерному корню вычислительных сетей. Первые компьютеры 50-х годов — большие, громоздкие и дорогие — предназначались для очень небольшого числа избранных пользователей. Часто эти монстры занимали целые здания. Такие компьютеры не были предназначены для интерактивной работы пользователя, а использовались в режиме пакетной обработки.

Системы пакетной обработки, как правило, строились на базе **мэйнфрейма** - мощного и надежного компьютера универсального назначения.

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

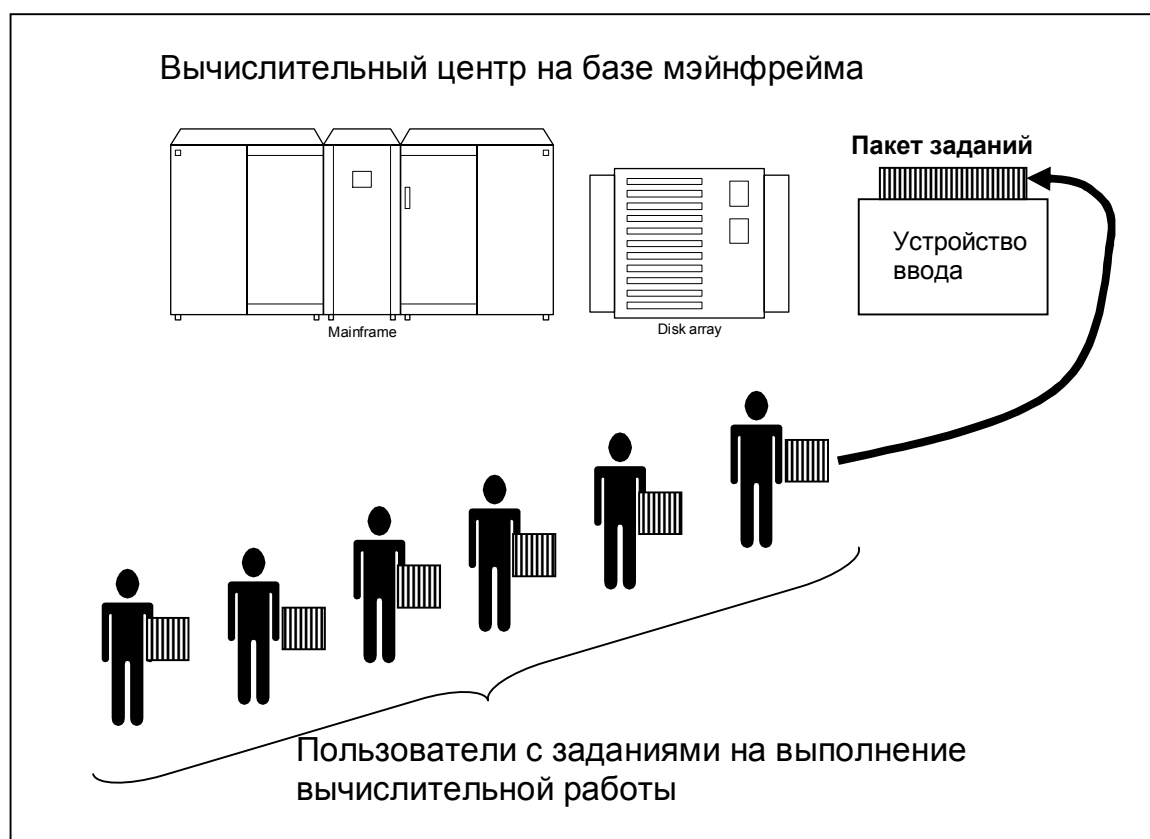


Рис. 1.2. Централизованная система на базе мэйнфрейма

Многотерминальные системы — прообраз сети

По мере удешевления процессоров в начале 60-х годов появились новые способы организации вычислительного процесса, которые позволили учесть интересы пользователей. Начали развиваться интерактивные **многотерминальные системы разделения времени** (рис. 1.3).

Терминал – устройство, соединенное с компьютером посредством линий связи и выполняющее определенные ограниченные действия.

В таких системах каждый пользователь получал собственный терминал, с помощью которого он мог вести диалог с компьютером. Количество одновременно работающих с компьютером пользователей зависело от его мощности так, чтобы время реакции вычислительной системы было достаточно мало, и пользователю была не слишком заметна параллельная работа с компьютером других пользователей.

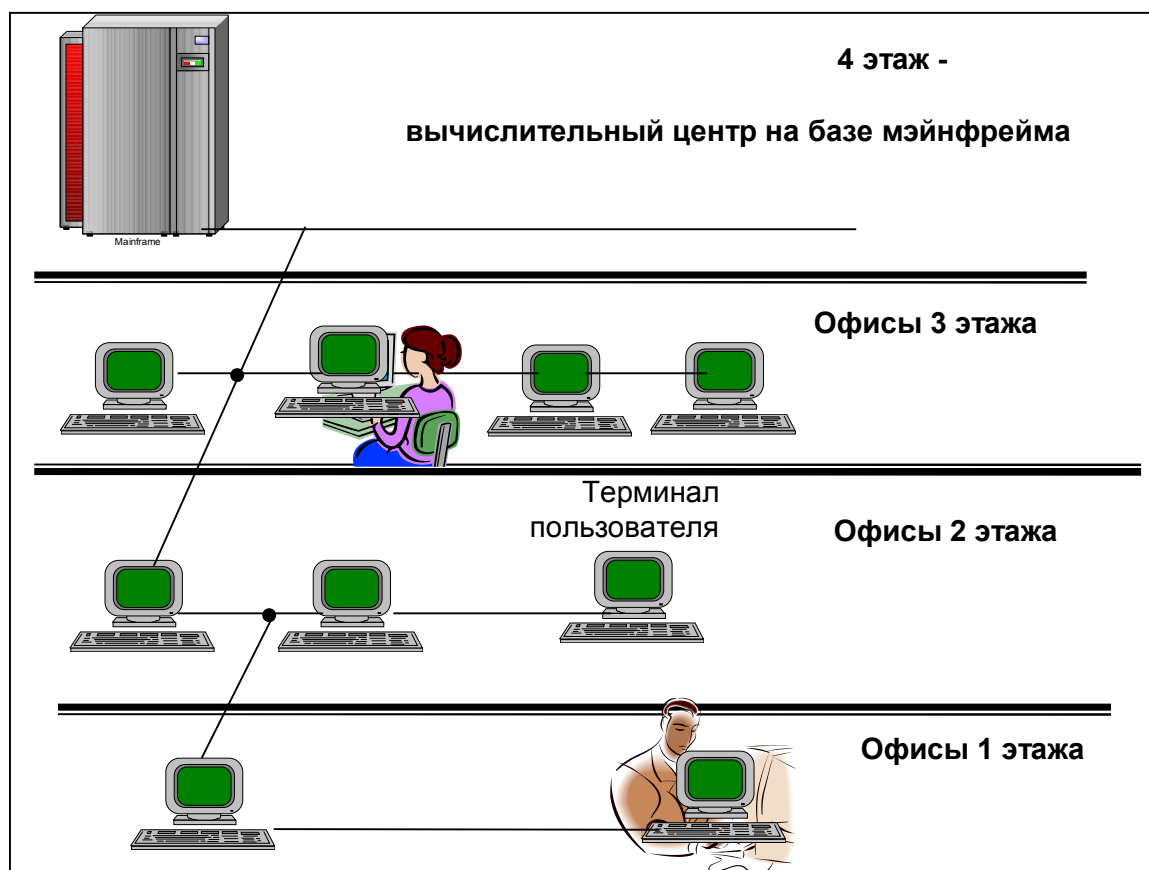


Рис. 1.3. Многотерминальная система — прообраз вычислительной сети

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

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

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

С другой стороны, и потребность предприятий в создании локальных сетей в это время еще не созрела — в одном здании просто нечего было объединять в сеть, так как из-за высокой стоимости вычислительной техники предприятия не могли себе позволить роскошь приобретения нескольких компьютеров. В этот период был справедлив так называемый **закон Гроша**, который эмпирически отражал уровень технологии того времени.

В соответствии с этим законом производительность компьютера была пропорциональна квадрату его стоимости, отсюда следовало, что за одну и ту же сумму было выгоднее купить одну мощную машину, чем две менее мощных — их суммарная мощность оказывалась намного ниже мощности дорогой машины.

Первые глобальные сети

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

Компьютеры получили возможность обмениваться данными в автоматическом режиме, что, собственно, и является базовым признаком любой вычислительной сети. На основе подобного механизма в первых сетях были реализованы службы обмена файлами, синхронизации баз данных, электронной почты и другие, ставшие теперь традиционными сетевые службы.

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

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

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

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

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

Пакет – блок данных стандартного формата. Формат пакета – заголовок, данные.

Так как прокладка высококачественных линий связи на большие расстояния обходится очень дорого, то в первых глобальных сетях часто использовались уже существующие каналы связи, изначально предназначенные совсем для других целей. Например, в течение многих лет глобальные сети строились на основе телефонных **каналов тональной частоты**, способных в каждый момент времени вести передачу только одного разговора в аналоговой форме.

Речевые сигналы имеют спектр шириной примерно в 10 000 Гц, однако основные гармоники укладываются в диапазон от 300 до 3400 Гц. Поэтому для качественной речи достаточно образовать между двумя собеседниками канал с полосой пропускания в 3100 Гц, который **называется каналом тональной частоты** и используется в телефонных сетях для соединения двух абонентов.

В цифровой телефонии для качественной передачи голоса необходима пропускная способность канала связи равная 64 Кб/с. Такой канал называется **элементарным каналом цифровых телефонных линий**.

Поскольку скорость передачи дискретных компьютерных данных по каналам тональной частоты была очень низкой, набор предоставляемых услуг в глобальных сетях такого типа обычно ограничивался передачей файлов, и электронной почтой. Помимо низкой скорости такие каналы имеют и другой недостаток - они вносят значительные искажения в передаваемые сигналы. Поэтому протоколы глобальных сетей, построенных с использованием каналов связи низкого качества, отличаются сложными процедурами контроля и восстановления данных. Типичным примером таких сетей являются сети X.25, разработанные еще в начале 70-х, когда низкоскоростные аналоговые каналы, арендуемые у телефонных компаний, были преобладающим типом каналов, соединяющих компьютеры и коммутаторы глобальной вычислительной сети.

Каналы низкого качества обеспечивают уровень ошибок 10^{-3} степени, каналы высокого качества – 10^{-9} степени.

В 1969 году Министерство обороны США инициировало работы по объединению в единую сеть суперкомпьютеров обороны и научно-исследовательских центров. Эта сеть, получившая название ARPANET, стала 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. ОС этих компьютеров стали предшественниками первых сетевых операционных систем.

Истинно **сетевые ОС** в отличие от **многотерминальных ОС** позволили 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 стороны, обладает некоторыми дополнительными средствами, позволяющими ей взаимодействовать через сеть с операционными системами других компьютеров. Программные модули, реализующие сетевые функции, появлялись в операционных системах постепенно, по мере развития сетевых технологий, аппаратной базы компьютеров и возникновения новых задач, требующих сетевой обработки.

Прогресс глобальных компьютерных сетей во многом определялся прогрессом телефонных сетей.

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

Это привело к появлению высокоскоростных цифровых каналов, соединяющих автоматические телефонные станции (АТС) и позволяющих одновременно передавать десятки и сотни разговоров. Была разработана специальная технология 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 других, так называемых **наложенных сетей**, которые уже работают на конечного пользователя.

См. рис. 1.3а

Сначала технология первичных сетей была исключительно внутренней технологией телефонных компаний. Однако со временем эти компании стали сдавать часть своих цифровых каналов, образованных в первичных сетях, в аренду предприятиям, которые использовали их для создания собственных телефонных и глобальных компьютерных сетей. Сегодня первичные сети обеспечивают скорости передачи данных до сотен гигабит (а в некоторых случаях до нескольких терабит) в секунду и густо покрывают территории всех развитых стран.

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

Первые локальные сети

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

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

Предприятие

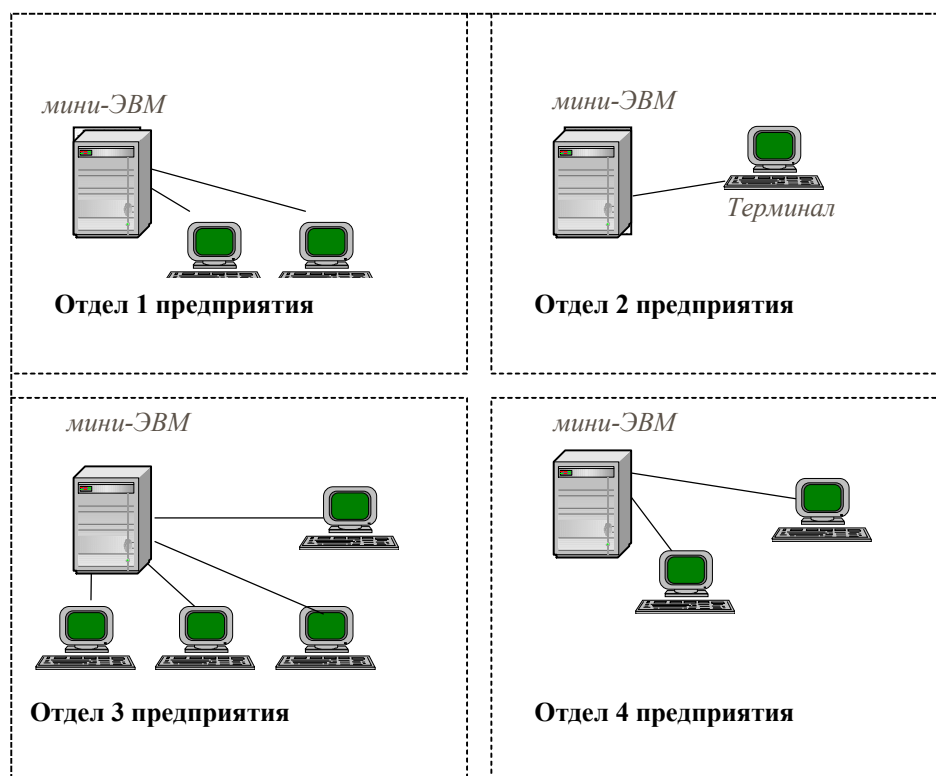


Рис. 1.4. Автономное использование нескольких мини-компьютеров на одном предприятии

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

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

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

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

Разнообразные устройства сопряжения, использующие собственные способы представления данных на линиях связи, свои типы кабелей и т. п., могли соединять только те конкретные модели компьютеров, для которых были разработаны, например, мини-компьютеры PDP-11 с мэйнфреймом IBM 360.

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

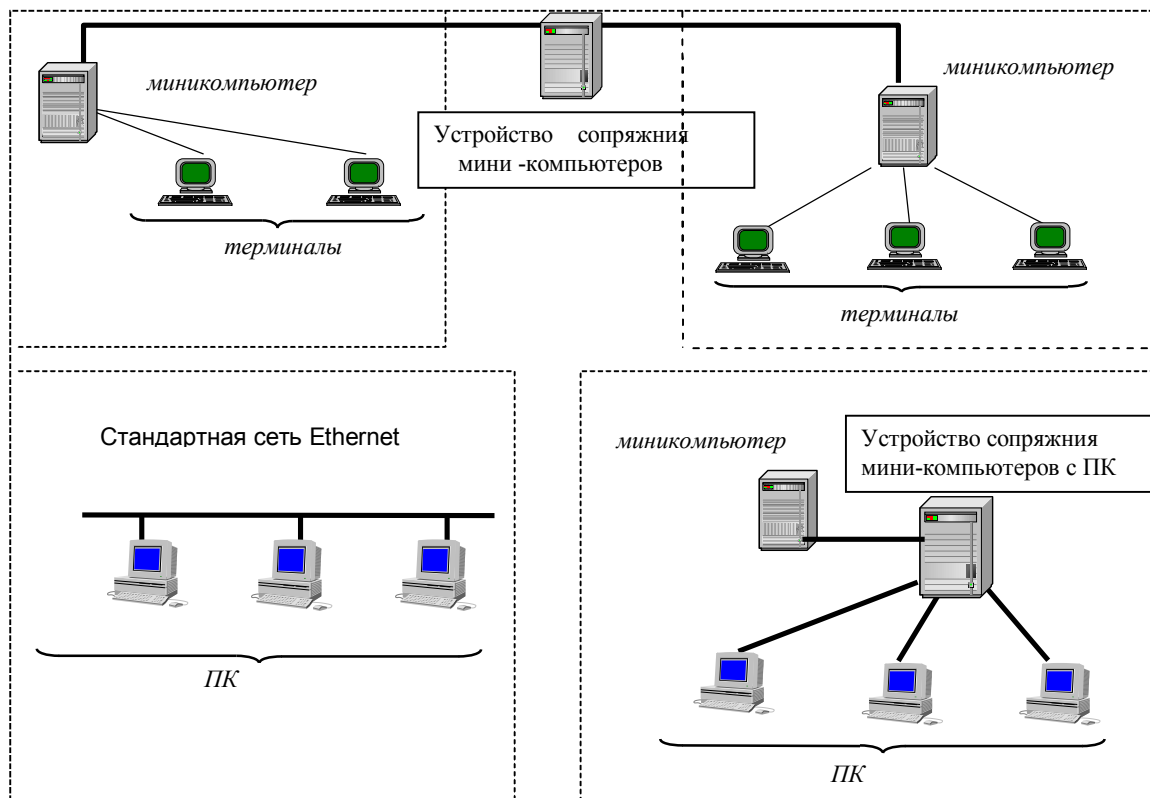


Рис. 1.5. Различные типы связей в первых локальных сетях

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

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

Стандартные сетевые технологии значительно упростили процесс построения локальной сети. Для создания сети достаточно было приобрести стандартный кабель, сетевые адаптеры соответствующего стандарта, например Ethernet, вставить адаптеры в компьютеры, присоединить их к кабелю стандартными разъемами.

Последствием и одновременно движущей силой прогресса локальных сетей стало появление огромного числа непрофессиональных пользователей, освобожденных от необходимости изучать специальные (и достаточно сложные) команды для сетевой работы.

Главным образом, это связано с использованием в локальных сетях качественных кабельных линий связи, на которых даже сетевые адаптеры первого поколения обеспечивали скорость передачи данных до 10 Мбит/с. При небольшой протяженности, свойственной локальным сетям, стоимость таких линий связи была вполне приемлемой. Поэтому экономное расходование пропускной способности каналов, одна из основных задач технологий первых глобальных сетей, никогда не выходило на первый план при разработке протоколов локальных сетей.

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

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

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

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

Этап	Время
Первые глобальные связи компьютеров, первые эксперименты с пакетными сетями	Конец 60-х
Начало передач по телефонным сетям голоса в цифровой форме	Конец 60-х
Появление больших интегральных схем. Первые мини-компьютеры. Первые нестандартные локальные сети	Начало 70-х
Стандартизация технологии X.25	1974
Появление персональных компьютеров, создание Интернета в современном виде, установка на всех узлах стека TCP/IP	Начало 80-х
Появление стандартных технологий локальных сетей (Ethernet - 1984 г., Token Ring - 1985 г., FDDI -1989 г)	Середина 80-х
Начало коммерческого использования Интернета	Конец 80-х
Изобретение Web	1991

Сближение локальных и глобальных сетей

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

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

Постепенно различия между локальными и глобальными типами сетевых технологий стали сглаживаться. Изолированные ранее локальные сети начали объединять друг с другом, при этом в качестве связующей среды использовались глобальные сети. Тесная интеграция локальных и глобальных сетей привела к значительному взаимопроникновению соответствующих технологий.

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

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

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

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

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

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

Еще одним признаком сближения локальных и глобальных сетей является появление сетей, занимающих промежуточное положение между локальными и глобальными сетями. **Городские сети**, или **сети мегаполисов** (Metropolitan Area Networks, MAN), предназначены для обслуживания территории крупного города.

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

Конвергенция компьютерных и телекоммуникационных сетей

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

К телекоммуникационным сетям относятся телефонные сети, радиосети и телевизионные сети. Главное, что объединяет их с компьютерными сетями, — то, что в качестве ресурса, предоставляемого клиентам, выступает **информация**. Однако эти сети, как правило, представляют информацию в разном виде. Так, изначально компьютерные сети разрабатывались для передачи алфавитно-цифровой информации, которую часто называют просто данными, в результате у компьютерных сетей имеется и другое название — **сети передачи данных**, в то время как телефонные сети и радиосети были созданы для передачи только голосовой информации, а телевизионные сети передают и голос, и изображение.

Несмотря на это, конвергенция телекоммуникационных и компьютерных сетей идет по нескольким направлениям.

Прежде всего, наблюдается *сближение видов услуг*, предоставляемых клиентам. Первая и не очень успешная попытка создания мультисервисной сети, способной оказывать различные услуги, в том числе услуги телефонии и передачи данных, привела к появлению технологии **цифровых сетей с интегрированным обслуживанием** (Integrated Services Digital Network, ISDN). Однако на практике ISDN предоставляет сегодня в основном телефонные услуги. Сегодня на роль глобальной **мультисервисной сети нового поколения**, претендует Интернет. Наибольшую привлекательность сейчас представляют собой новые виды комбинированных услуг, в которых сочетаются несколько традиционных услуг, например, услуга **универсальной**

службы сообщений, объединяющей электронную почту, телефонию, факсимильную службу и пейджинговую связь. Наибольших успехов на практическом поприще достигла IP-телефония, услугами которой прямо или косвенно сегодня пользуются миллионы людей. Однако для того, чтобы стать мультисервисной сетью, Интернету еще предстоит пройти большой путь.

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

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

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

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

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

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

Сегодня становится все более очевидным, что мультисервисная сеть нового поколения не может быть создана в результате «победы» какой-нибудь одной технологии или подхода. Ее может породить только процесс конвергенции, когда от каждой технологии будет взято все самое лучшее и соединено в некоторый новый сплав, который и даст требуемое качество для поддержки существующих и создания новых услуг. Взамен термина **мультисервисная сеть** появился новый термин — **инфокоммуникационная сеть**, который прямо говорит о двух составляющих современной сети - информационной (компьютерной) и телекоммуникационной.

Выводы

Компьютерные сети стали логическим результатом эволюции компьютерных и телекоммуникационных технологий. С одной стороны, они являются частным случаем

распределенных компьютерных систем, а с другой стороны, могут рассматриваться как средство передачи информации на большие расстояния, для чего в них применяются методы кодирования и мультиплексирования данных, получившие развитие в различных телекоммуникационных системах.

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

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

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

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

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

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

Тенденция сближения различных типов сетей характерна не только для локальных и глобальных компьютерных сетей, но и для телекоммуникационных сетей других типов: телефонных сетей, радиосетей, телевизионных сетей. В настоящее время ведутся активные работы по созданию универсальных мультисервисных сетей, способных одинаково эффективно передавать информацию любого типа: данные, голос и видео.

Вопросы и задания

1. Какие свойства многотерминальной системы отличают ее от компьютерной сети?
2. Когда впервые были получены значимые практические результаты по объединению компьютеров с помощью глобальных связей?
3. Что такое ARPANET?
4. Какое из этих событий произошло позже других:
 - изобретение Web;
 - появление стандартных технологии LAN;
 - начало передачи голоса в цифровой форме по телефонным сетям.
5. Какое событие послужило стимулом к активизации работ по созданию LAN?
6. Когда была стандартизована технология Ethernet? Token Ring? FDDI?
7. По каким направлениям идет сближение компьютерных и телекоммуникационных сетей.

8. Поясните термины «мультисервисная сеть», «инфокоммуникационная сеть», «интеллектуальная сеть».
9. Поясните, почему сети WAN появились раньше, чем сети LAN.

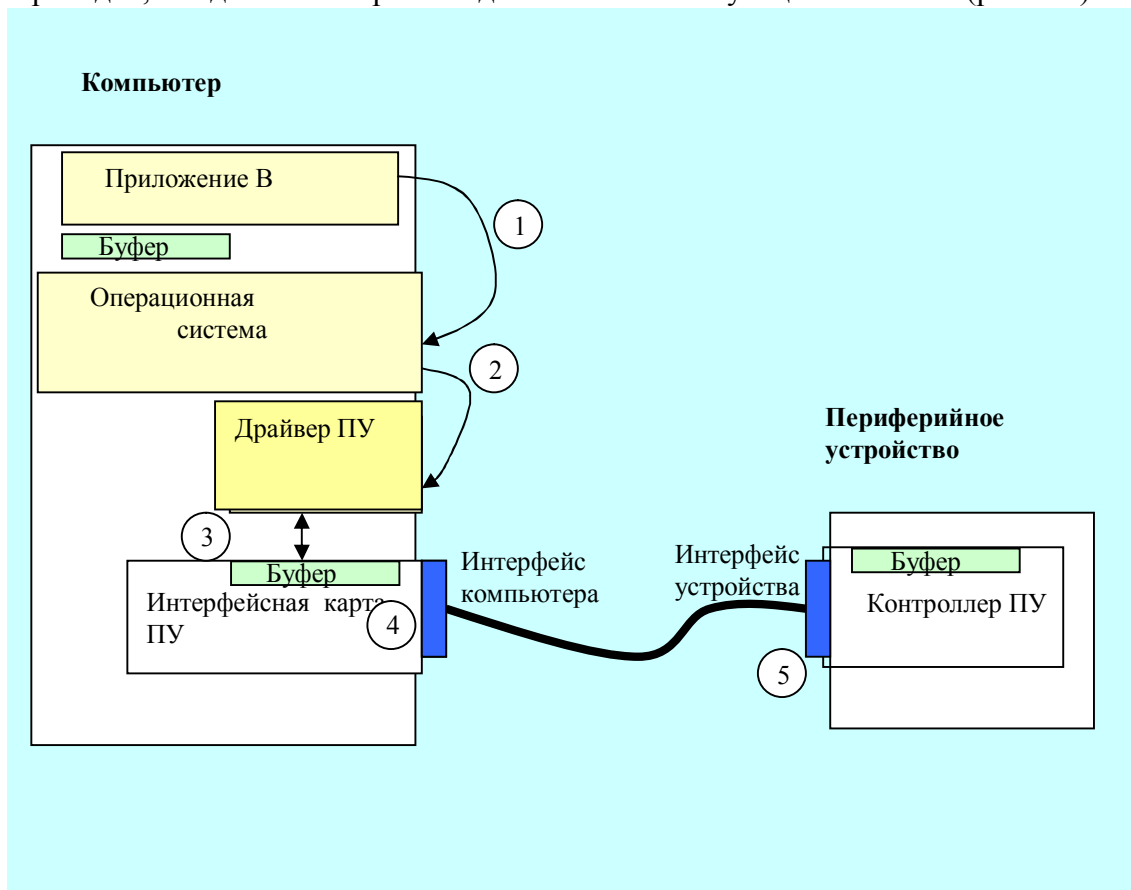
Связь компьютера с периферийными устройствами

Для организации связи между компьютером и периферийным устройством (ПУ) в обоих этих устройствах предусмотрены внешние физические интерфейсы.

Интерфейс – в широком смысле – формально определенная логическая и физическая границы между взаимодействующими независимыми объектами. Интерфейс задает параметры, процедуры и характеристики взаимодействия объектов

Физический интерфейс (называемый также **портом**) - определяется набором электрических связей и характеристиками сигналов.

Обычно он представляет собой разъем с набором контактов, каждый из которых имеет определенное назначение, например, это может быть группа контактов для передачи данных, контакт синхронизации данных и т. п. Пара разъемов соединяется кабелем, который состоит из набора проводов, каждый из которых соединяет соответствующие контакты (рис. 2.2).



2.2. Взаимодействие компьютера с периферийным устройством. 1- запрос приложения к ОС, 2 – вызов драйвера, 3 – загрузка в буфер интерфейсной карты команды или данных, 4 – побитная передача информации в линию связи, 5 - прием битов и размещение их в буфере

Логический интерфейс — это набор информационных сообщений определенного формата, которыми обмениваются два устройства или две программы, а также набор правил, определяющих логику обмена этими сообщениями.

Примерами стандартных интерфейсов, используемых в компьютерах, являются параллельный (передающий данные байтами) интерфейс Centronics, предназначенный, как правило, для подключения принтеров, и последовательный интерфейс (передающий данные битами) RS-232C (известный также как COM-порт), который имеет более универсальное назначение — он поддерживается не только принтерами, но и графопостроителями, манипуляторами типа «мышь» и многими другими устройствами. Существуют также специализированные интерфейсы, которые предназначены для подключения уникальных периферийных устройств, например сложной физической экспериментальной установки. В настоящее время широкое распространение нашел интерфейс **USB** (Universal Serial Bus - универсальная последовательная шина).

В компьютере операции интерфейса реализуются совокупностью аппаратных и программных средств: интерфейсной платой (аппаратное устройство) и специальной программой, управляющей этим контроллером, которую называют драйвером соответствующего периферийного устройства.

В ПУ интерфейс чаще всего полностью реализуется аппаратным устройством - **контроллером**, хотя встречаются и программно-управляемые контроллеры для управления современными принтерами, обладающими более сложной логикой.

Периферийные устройства могут принимать от компьютера как данные, например байты информации, которую нужно распечатать на бумаге, так и команды управления, в ответ на которые контроллер ПУ может выполнять специальные действия (перевести головку диска на требуемую дорожку, вытолкнуть лист бумаги из принтера и т. д.). Контроллер принтера, например, поддерживает некоторый набор достаточно простых команд, таких как «Печать символа», «Перевод строки», «Возврат каретки» и т. п., которые он получает от компьютера по интерфейсу и обрабатывает, управляя электромеханическими частями принтера.

Итак, рассмотрим порядок действий, в результате которых приложение распечатывает данные на принтере.

1. Приложение обращается с запросом на выполнение операции ввода-вывода к операционной системе. В запросе указываются адрес данных в оперативной памяти, идентифицирующая информация о периферийном устройстве и операция, которую надо выполнить.

2. Получив запрос, операционная система запускает драйвер принтера. Дальнейшие действия по выполнению операции ввода-вывода со стороны компьютера реализуются интерфейсной картой, работающей под управлением драйвера.

3. Драйвер принтера оперирует командами, понятными контроллеру принтера, то есть командами «Печать символа», «Перевод строки», «Возврат каретки». Драйвер в определенной последовательности помещает коды этих команд в регистр интерфейсной карты, которая побайтно передает их по линиям связи контроллеру периферийного устройства. Для одного и тот же контроллера можно разработать различные драйверы, которые с помощью одного набора команд будут реализовывать разные алгоритмы управления ПУ.

4. Интерфейсная карта выполняет низкоуровневую работу, она не вдастся в смысл данных и команд, передаваемых ей драйвером, считая их однородным потоком битов. После получения от драйвера очередного байта интерфейсная карта просто последовательно передает биты в линию связи, представляя каждый бит электрическим сигналом.

контроллер COM-порт работает под управлением драйвера COM-порта. Вместе они обеспечивают передачу по кабелю между компьютерами одного байта информации.

В «настоящих» локальных сетях подобные функции передачи данных в линию связи выполняются сетевыми интерфейсными картами (Network Interface Card, NIC), называемыми также сетевыми адаптерами, и их драйверами.

Итак, механизм обмена байтами между двумя компьютерами определен. Однако этого еще недостаточно для решения поставленной задачи — распечатки текста на «чужом» принтере. В частности, необходимо, чтобы компьютер В «понял», какую операцию он должен выполнить с передаваемыми данными, на каком из имеющихся в его распоряжении устройств, в каком виде должен быть распечатан текст и т. п. Обо всем этом должны договориться приложения А и В путем обмена сообщениями.

Чтобы приложения могли «понимать» получаемую друг от друга информацию, программисты, разрабатывавшие приложения А и В, должны *строго оговорить* форматы сообщений, которыми будут обмениваться приложения, и их семантику. Например, они могут договориться о том, что любое выполнение удаленной операции печати начинается с передачи сообщения, запрашивающего информацию о готовности приложения В; что в следующем сообщении идут идентификаторы компьютера и пользователя, сделавшего запрос; что признаком срочного завершения печати является определенная кодовая комбинация и т. п. **Тем самым определяется протокол взаимодействия приложений.**

Протокол — это набор информационных сообщений определенного формата, которыми обмениваются два устройства или две программы, а также набор правил, определяющих логику обмена этими сообщениями.

Рассмотрим взаимодействие всех элементов этой небольшой сети, которые позволят приложению на компьютере А распечатать текст на принтере компьютера В

1. Приложение А формирует сообщение-запрос для приложения В на печать текста и помещает его в свой буфер. Чтобы передать данный запрос компьютеру В, приложение А обращается к локальной ОС, которая запускает драйвер COM-порта компьютера и сообщает ему адрес буфера, где хранится запрос. Затем, по ранее описанной схеме, драйвер и контроллер COM-порта компьютера А, взаимодействуя с драйвером и контроллером COM-порта компьютера В, передают сообщение байт за байтом в компьютер В.

2. Драйвер COM-порта компьютера В постоянно находится в режиме ожидания прихода информации из внешнего мира. В некоторых случаях драйвер вызывается асинхронно, по прерываниям от контроллера. Получив очередной байт и убедившись в его корректности, драйвер помещает его в буфер приложения В.

3. Приложение В принимает сообщение, интерпретирует его и формирует запрос к локальной ОС на выполнение тех или иных действий с принтером. В ходе печати могут возникнуть ситуации, о которых необходимо сообщить приложению А. В этом случае используется симметричная схема: теперь запрос на передачу сообщения поступает от приложения В к локальной ОС компьютера В. Драйверы и контроллеры COM-портов обоих компьютеров организуют побайтную передачу сообщения, которое затем помещается в буфер приложения А.

Потребность в доступе к удаленным файлам может возникать у пользователей многих других приложений: текстового редактора, графического редактора, системы управления базой данных (СУБД). Очевидно, нерационально включать рассмотренные универсальные функции по организации ввода-вывода в состав каждого приложения. Более эффективно решают задачу пара специализированных программных модулей.

Клиент – системный программный модуль, предназначенный для формирования сообщений-запросов к удаленной машине от разных приложений, а затем приема результатов и передачи их соответствующим приложениям.

Сервер – системный программный модуль, который постоянно ожидает прихода по сети запросов от клиентов и, приняв запрос, пытается его выполнить, возможно, с участием локальной ОС. Один сервер может выполнять запросы сразу нескольких клиентов (последовательно или одновременно)

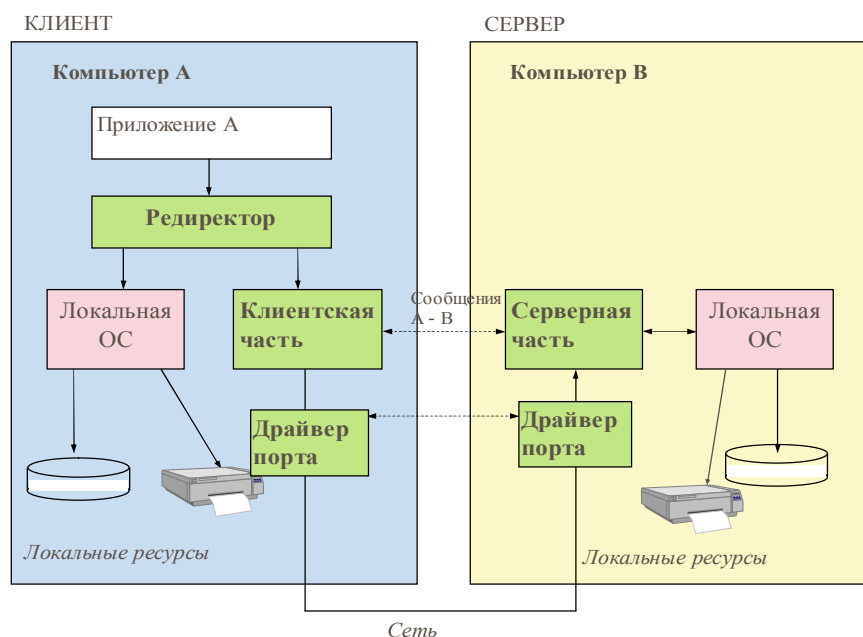


Рис. 2.4. Взаимодействие программных компонентов при связи двух компьютеров

Очень удобной и полезной функцией клиентской программы является способность отличить запрос к **удаленному** ресурсу от запроса к **локальному** ресурсу. Если клиентская программа умеет это делать, то приложения не должны заботиться о том, с каким принтером они работают (локальным или удаленным), клиентская программа сама распознает и **перенаправляет** (redirect) запрос к удаленной машине. Отсюда и название, часто используемое для клиентского модуля, — **редиректор**.

Клиент и сервер выполняют системные функции по обслуживанию запросов всех приложений компьютера А на удаленный доступ к ресурсу (принтеру, файлам, факсу) компьютера В. Чтобы приложения компьютера В могли пользоваться ресурсами компьютера А, описанную схему нужно симметрично дополнить клиентом для компьютера В и сервером для компьютера А.

Схема взаимодействия клиента и сервера с приложениями и локальной операционной системой приведена на **рис. 2.4**.

Взаимодействие между компьютерами сети происходит за счет передачи сообщений через сетевые адаптеры и каналы связи. С помощью этих сообщений один компьютер обычно запрашивает доступ к локальным ресурсам другого компьютера. Такими ресурсами могут быть как данные, хранящиеся на диске, так и разнообразные периферийные устройства – принтеры, плоттеры, факс-аппараты и т.д.

Разделение локальных ресурсов каждого компьютера между всеми пользователями сети – основная цель создания вычислительной сети.

Несмотря на то, что мы рассмотрели очень простую схему связи только двух компьютеров, функции программ, обеспечивающих удаленный доступ к принтеру, во многом совпадают с функциями сетевой операционной системы, работающей в сети с более сложными аппаратными связями компьютеров.

К основным функциональным компонентам сетевой ОС относятся **средства управления локальными ресурсами и сетевые средства**.

Сетевые средства можно разделить на три компоненты:

1. Средства предоставления локальных ресурсов и услуг в общее пользование – серверная часть ОС.
2. Средства запроса доступа к удаленным ресурсам – клиентская часть ОС.
3. Транспортные средства ОС, которые с коммуникационной системой обеспечивают передачу сообщений между компьютерами сети.

В связи с этим, локальные сети можно подразделить в зависимости от используемой сетевой операционной системы на:

- **Серверные** сети
- **Одноранговые** сети
- **Комбинированные** сети.

Термины «клиент» и «сервер» используются для обозначения не только программных модулей, но и компьютеров, подключенных к сети. Если компьютер преимущественно предоставляет свои ресурсы другим компьютерам сети, то он называется **сервером**, а если он их потребляет — **клиентом**. Иногда один и тот же компьютер может одновременно играть роли и сервера, и клиента.

Сетевые службы и приложения

Предоставление пользователям совместного доступа к определенному типу ресурсов, например к файлам, называют также предоставлением сервиса.

Обычно сетевая операционная система поддерживает несколько видов сетевых сервисов для своих пользователей — файловый сервис, сервис печати, сервис электронной почты, сервис удаленного доступа и т. п. Программы, реализующие сетевые сервисы, относятся к классу распределенных программ.

Распределенная программа – это программа, которая состоит из нескольких взаимодействующих частей (в приведенном на **рис. 2.5** примере – из двух), причем каждая часть, как правило, выполняется на отдельном компьютере сети.



Рис. 2.5. Взаимодействие частей распределенного приложения

Сетевые службы – это *системные* распределенные программы, реализующие сетевые сервисы. Они представляют собой пару «клиент-сервер» и являются неотъемлемыми компонентами сетевой ОС.

Однако в сети могут выполняться и распределенные пользовательские приложения. **Распределенное приложение также состоит из нескольких частей, каждая из которых выполняет какую-то определенную законченную работу по решению прикладной задачи.** Например, одна часть приложения, выполняющаяся на компьютере пользователя, может поддерживать специализированный графический интерфейс, вторая — работать на мощном выделенном компьютере и заниматься статистической обработкой введенных пользователем данных, третья — заносить полученные результаты в базу данных на компьютере с установленной стандартной СУБД. Распределенные приложения в полной мере используют потенциальные возможности распределенной обработки, предоставляемые вычислительной сетью, и поэтому часто называются **сетевыми приложениями**.

Не всякое приложение, выполняемое в сети, является распределенным. Значительная часть истории локальных сетей связана как раз с использованием таких нераспределенных приложений. Рассмотрим, например, как происходила работа пользователя с известной в свое время СУБД dBase. Файлы базы данных, с которыми работали все пользователи сети, располагались на файловом сервере. Сама же СУБД хранилась на каждом клиентском компьютере в виде единого программного модуля. Программа dBase была рассчитана только на обработку данных, расположенных на том же компьютере, что и сама программа. Пользователь запускал dBase на своем компьютере, и программа искала данные на локальном диске, совершенно не принимая во внимание существование сети. Чтобы обрабатывать с помощью dBase данные, расположенные на удаленном компьютере, пользователь обращался к услугам файловой службы, которая доставляла данные с сервера на клиентский компьютер и создавала для СУБД эффект их локального хранения.

Большинство приложений, используемых в компьютерных сетях в середине 80-х годов, являлись обычными нераспределенными приложениями. И это понятно — они были написаны для автономных компьютеров, а потом просто были перенесены в сетевую среду. Создание же распределенных приложений, хотя и сулило много преимуществ (снижение сетевого трафика, специализация компьютеров), оказалось делом совсем не простым. Нужно было решать множество дополнительных проблем:

- на сколько частей разбить приложение,
- какие функции возложить на каждую часть,
- как организовать взаимодействие этих частей, чтобы в случае сбоев и отказов оставшиеся части корректно завершали работу и т. д.

Поэтому до сих пор только небольшая часть приложений являются распределенными, хотя очевидно, что именно за этим классом приложений будущее, так как они в полной мере могут использовать потенциальные возможности сетей, по распараллеливанию вычислений.

Физическая передача данных по линиям связи

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

Кодирование

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

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

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

Аналогичные подходы применимы для кодирования данных и при передаче их между двумя компьютерами **по линиям связи**. Однако эти линии связи отличаются по своим характеристикам от линий внутри компьютера. **Главное отличие** внешних линий связи от внутренних состоит в их гораздо большей протяженности, а также в том, что они проходят вне экранированного корпуса по пространствам, зачастую подверженным воздействию сильных электромагнитных помех. Все это приводит к существенно большим искажениям прямоугольных импульсов (например, **«заваливанию» фронтов**), чем внутри компьютера. Поэтому для надежного распознавания импульсов на приемном конце линии связи при передаче данных внутри и вне компьютера не всегда можно использовать одни и те же скорости и способы кодирования. Например, медленное нарастание фронта импульса из-за высокой емкостной нагрузки линии требует, чтобы импульсы передавались с меньшей скоростью (чтобы передний и задний фронты соседних импульсов не перекрывались, и импульс успел «дорости» до требуемого уровня).

В вычислительных сетях применяют как потенциальное, так и импульсное кодирование дискретных данных, а также специфический способ представления данных, который никогда не используется внутри компьютера, — **модуляцию** (рис. 2.6). При модуляции дискретная информация представляется синусоидальным сигналом той частоты, которую хорошо перелает имеющаяся линия связи.

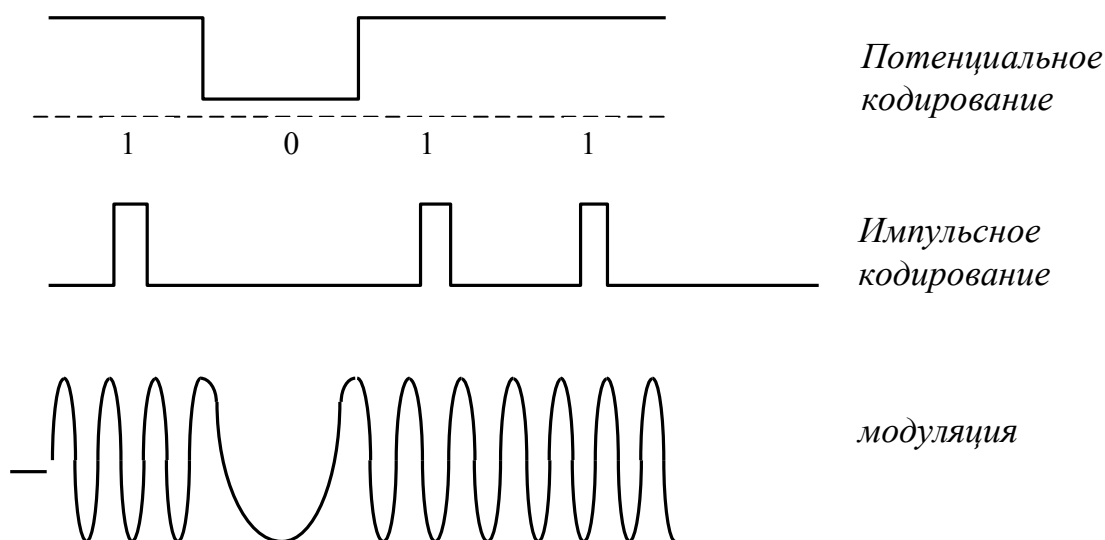


Рис. 2.6. Примеры представления дискретной информации

Модуляция является таким способом физического кодирования при котором информация кодируется изменением амплитуды, частоты или фазы синусоидального сигнала несущей частоты. (Рис. 2.6а)

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

При **частотной модуляции** значения 0 и 1 исходных данных передаются синусоидами с различной частотой – f_0 и f_1 . Этот способ модуляции не требует сложных схем в модемах и обычно применяется в низкоскоростных модемах, работающих на скоростях 300 или 1200 бит/с.

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

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

На способ передачи сигналов влияет и *количество проводов* в линиях связи между компьютерами. Для снижения стоимости линий связи в сетях обычно стремятся к сокращению количества проводов и из-за этого используют не параллельную передачу всех битов одного байта или даже нескольких байтов, как это делается внутри компьютера, а последовательную побитную передачу, требующую всего одной пары проводов.

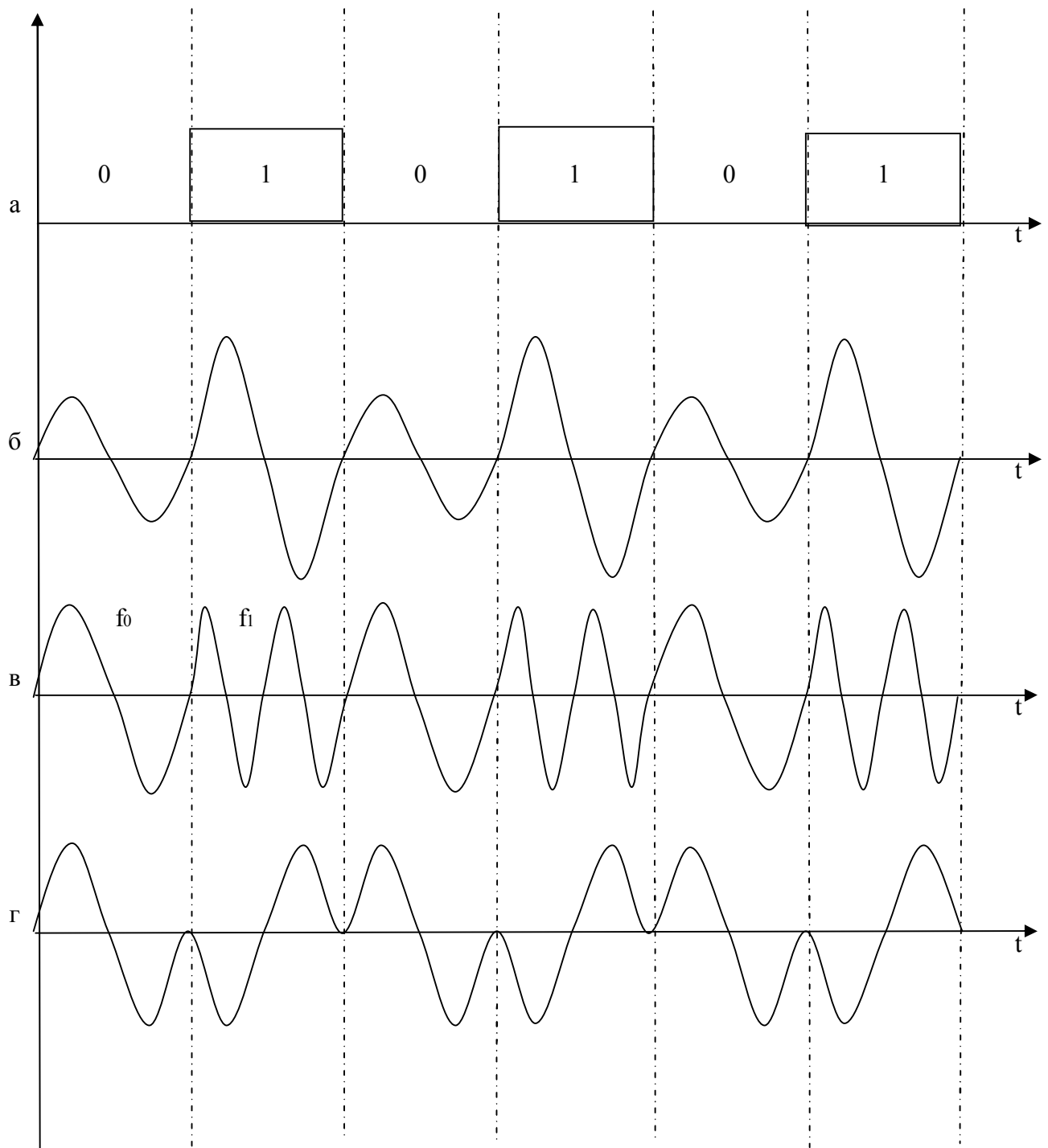


Рис. 2.6а. Различные типы модуляций

Еще одной проблемой, которую нужно решать при передаче сигналов, является проблема взаимной **синхронизации** передатчика одного компьютера с приемником другого.

Синхронизация передатчика и приемника нужна для того, чтобы приемник точно знал, в какой момент времени необходимо считывать новую информацию с линии связи. Эта проблема в сетях решается сложнее, чем при обмене данными между близко расположенными

устройствами, например между блоками внутри компьютера или же между компьютером и принтером.

При связи компьютера с периферийным устройством синхронизация обеспечивается путем периодической синхронизации заранее обусловленными кодами или импульсами характерной формы, отличающейся от формы импульсов данных.

На небольших расстояниях (внутри компьютера) хорошо работает схема, основанная на отдельной тактирующей линии связи (**рис. 2.6б**), так что информация снимается с линии данных только в момент прихода тактового импульса. В сетях использование этой схемы вызывает трудности из-за неоднородности характеристик проводников в кабелях.

На больших расстояниях неравномерность скорости распространения сигнала может привести к тому, что тактовый импульс придет настолько позже или раньше соответствующего сигнала данных, что бит данных будет пропущен или считан повторно. Другой причиной, по которой в сетях отказываются от использования тактирующих импульсов, является экономия проводников в дорогостоящих кабелях.

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

При использовании синусоид в качестве несущего сигнала результирующий код обладает свойством самосинхронизации, так как изменение амплитуды несущей частоты дает возможность приемнику определить момент появления входного кода.

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

Характеристики физических каналов

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

- **Предложенная нагрузка** - это поток данных, поступающий от пользователя на вход сети. Предложенную нагрузку можно характеризовать скоростью поступления данных в сеть — в битах в секунду (или килобитах, мегабитах и т. д.).
- **Скорость передачи данных** (information rate или throughput, оба английских термина используются равноправно) — это *фактическая* скорость потока данных, прошедшего через сеть. Эта скорость может быть меньше, чем скорость предложенной нагрузки, так как данные в сети могут искажаться или теряться.
- **Емкость канала связи** (capacity), называемая также **пропускной способностью**, представляет собой *максимально возможную* скорость передачи информации по каналу.

Спецификой этой характеристики является то, что она отражает не только параметры *физической среды передачи*, но и особенности *выбранного способа передачи* дискретной информации по этой среде. Например, емкость канала связи в сети Ethernet на оптическом волокне равна 10 Мбит/с. Эта скорость является

предельно возможной для сочетания технологии Ethernet и оптического волокна. Однако для того же самого оптического волокна можно разработать и другую технологию передачи данных, отличающуюся способом кодирования данных, тактовой частотой и другими параметрами, которая будет иметь другую емкость. Так, технология Fast Ethernet обеспечивает передачу данных по тому же оптическому волокну с максимальной скоростью 100 Мбит/с, а технология Gigabit Ethernet - 1000 Мбит/с. Передатчик коммуникационного устройства должен работать со скоростью, равной пропускной способности канала. Эта скорость иногда называется битовой скоростью передатчика (bit rate of transmitter).

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

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

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

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

Проблемы связи нескольких компьютеров

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

Топология физических связей

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

Под **топологией сети** понимается геометрическая фигура, вершинами которого соответствуют конечные узлы сети (например, компьютеры) и коммуникационное оборудование (например, маршрутизаторы), а ребрам – физические или информационные связи между вершинами

Виды коммуникационного оборудования, используемого в компьютерных сетях: **повторители, концентраторы, мосты, коммутаторы, маршрутизаторы, шлюзы.**

Число возможных вариантов конфигураций резко возрастает при увеличении числа связываемых устройств. Так, если три компьютера мы можем связать двумя способами (рис. 2.7, а), то для четырех можно предложить уже шесть топологически разных конфигураций (при условии неразличимости компьютеров), что и иллюстрирует **рис. 2.7, б.**

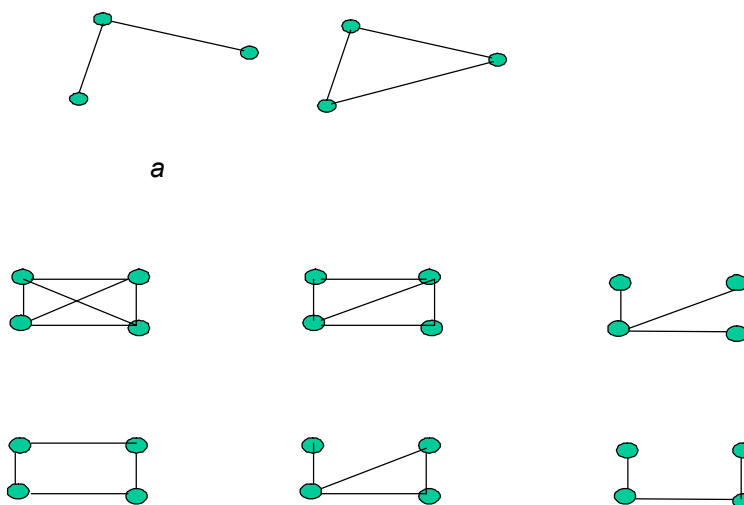


Рис. 2.7. Варианты связи компьютеров

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

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

Среди множества возможных конфигураций различают **полносвязные и неполносвязные.**

Полносвязная топология (рис. 2.8. а) соответствует сети, в которой каждый компьютер непосредственно связан со всеми остальными. Несмотря на логическую простоту, этот вариант оказывается громоздким и неэффективным. Действительно, в таком случае каждый компьютер в сети должен иметь большое количество коммуникационных портов, достаточное для связи с каждым из остальных компьютеров сети. Для каждой пары компьютеров должна быть выделена отдельная физическая линия связи. (В некоторых случаях даже две, если невозможно использование этой линии для двусторонней передачи.) Полносвязные топологии в

компьютерных сетях применяются редко. Чаше этот вид топологии используется в многомашинных комплексах или в глобальных сетях, объединяющих небольшое количество компьютеров.

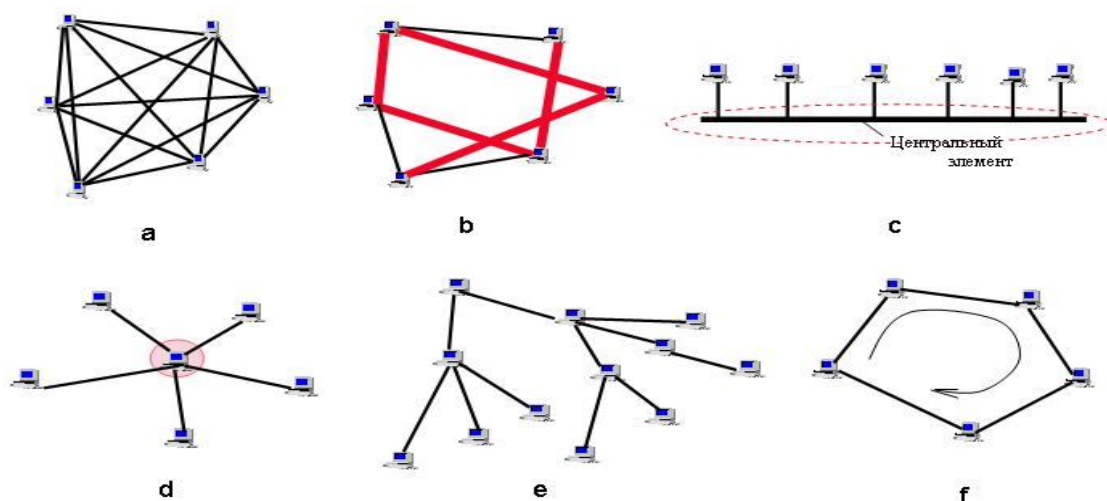


Рис. 2.8. Типовые топологии сетей

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

Ячеистая топология получается из полносвязной путем удаления некоторых связей (рис. 2.8. b). Ячеистая топология допускает соединение большого количества компьютеров и характерна, как правило, для крупных глобальных сетей.

В сетях с **кольцевой топологией** (рис. 2.8. f) данные передаются по кольцу от одного компьютера к другому. Кольцо представляет собой очень удобную конфигурацию и для организации обратной связи — данные, сделав полный оборот, возвращаются к узлу-источнику. Поэтому источник может контролировать процесс доставки данных адресату. В то же время в сетях с кольцевой топологией необходимо принимать специальные меры, чтобы в случае выхода из строя или отключения какого-либо компьютера не прерывался канал связи между остальными узлами кольца.

Звездообразная топология (рис. 2.8, d) образуется в случае, когда каждый компьютер подключается непосредственно к общему центральному устройству, называемому **концентратором**. В функции концентратора входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети. В качестве концентратора может выступать как универсальный компьютер, так и специализированное устройство. К недостаткам топологам типа звезда относится более высокая стоимость сетевого оборудования из-за необходимости приобретения специализированного центрального устройства. Кроме того, возможности по наращиванию количества узлов в сети ограничиваются количеством портов концентратора.

Иногда имеет смысл строить сеть с использованием нескольких концентраторов, иерархически соединенных между собой связями типа звезда (рис. 2.8, e). Получаемую в результате структуру называют **иерархической звездой**, а также **деревом**. В настоящее время дерево является самой распространенной топологией связей как в локальных, так и глобальных сетях.

Особым частным случаем звезды является конфигурация **общая шина** (рис. 2.8. с). Здесь в качестве центрального элемента выступает пассивный кабель, к которому по схеме «монтажного ИЛИ» подключается несколько компьютеров (такую же топологию имеют многие сети, использующие беспроводную связь — роль общей шины здесь играет общая радиосреда). Передаваемая информация распространяется по кабелю и доступна одновременно всем компьютерам, присоединенным к этому кабелю. Основными преимуществами такой схемы являются ее дешевизна и простота присоединения новых узлов к сети, а недостатками — низкая надежность (любой дефект кабеля полностью парализует всю сеть) и невысокая производительность (в каждый момент времени только один компьютер может передавать данные по сети, поэтому пропускная способность делится здесь между всеми узлами сети).

В то время как небольшие сети, как правило, имеют типовую топологию - звезда, кольцо или общая шина, для крупных сетей характерно наличие произвольных связей между компьютерами. В таких сетях можно выделить отдельные произвольно связанные фрагменты (подсети), имеющие типовую топологию, поэтому их называют сетями со **смешанной топологией** (рис. 2.9).

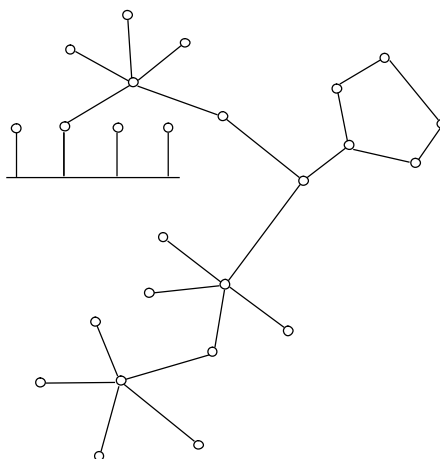


Рис. 2.9. Смешанная топология

Адресация узлов сети

Еще одной новой проблемой, которую нужно учитывать при объединении трех и более компьютеров, является проблема их адресации, точнее адресации их сетевых интерфейсов¹. Один компьютер может иметь несколько сетевых интерфейсов. Например, для создания полносвязной структуры из N компьютеров необходимо, чтобы у каждого из них имелся $N-1$ интерфейс. По количеству адресуемых интерфейсов адреса можно классифицировать следующим образом:

- **уникальный адрес** (unicast) используется для идентификации отдельных интерфейсов;
- **групповой адрес** (multicast) идентифицирует сразу несколько интерфейсов, поэтому данные, помеченные групповым адресом, доставляются каждому из узлов, входящих в группу;
- данные, направленные по **широковещательному адресу** (broadcast), должны быть доставлены всем узлам сети;
- 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 **числовыми** (например, 129.26.255.255 или 81.la.ff.ff) и **символьными** (site.domen.ru, willi-winki).

Символьные адреса (имена) предназначены для запоминания людьми и поэтому обычно несут смысловую нагрузку. Для работы в больших сетях символьное имя может иметь иерархическую структуру, например ftp-arch1.ucl.ac.uk. Хотя символьные имена удобны для людей, из-за переменного формата и потенциально большой длины их передача по сети не очень экономична.

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

Адресное пространство может иметь **плоскую (линейную)** организацию (рис. 2.10) или **иерархическую** организацию (рис. 2.11).

При плоской организации множество адресов никак не структурировано. Примером плоского числового адреса является MAC-адрес, предназначенный для однозначной идентификации сетевых интерфейсов в локальных сетях. Такой адрес обычно используется только аппаратурой, поэтому его стараются сделать по возможности компактным и записывают в виде двоичного или шестнадцатеричного числа, например 0081005e24a8. При задании MAC-адресов не требуется выполнение ручной работы, так как они обычно встраиваются к аппаратуру компанией - изготовителем, поэтому их называют также **аппаратными адресами** (hardware addresses). Использование плоских адресов является жестким решением — при замене аппаратуры, например, сетевого адаптера, изменяется и адрес сетевого интерфейса компьютера.

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

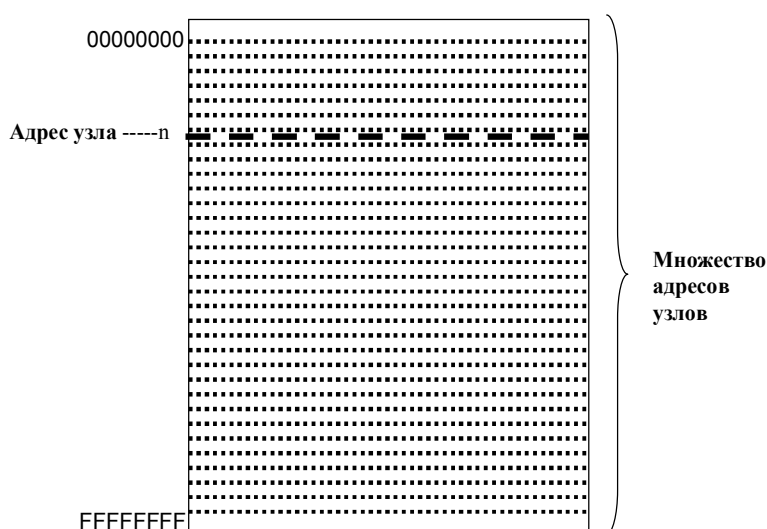


Рис. 2.10. Плоская организация адресного пространства



Рис. 2.11. Иерархическая структура 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. 2.11 трехуровневой структуре адресного пространства адрес конечного узла задается тремя составляющими: идентификатором группы (K), в которую входит данный узел, идентификатором подгруппы (L) и, наконец, идентификатором узла (n), однозначно определяющим его в подгруппе. Иерархическая адресация во многих случаях оказывается более рациональной, чем плоская. В больших сетях, состоящих из многих тысяч узлов, использование плоских адресов приводит к большим издержкам - конечным узлам и коммуникационному оборудованию приходится оперировать таблицами адресов, состоящими из тысяч записей. В противоположность этому иерархическая система адресации позволяет при перемещении данных до определенного момента пользоваться только старшей составляющей адреса (например, идентификатором группы K), затем для дальнейшей локализации адресата задействовать следующую по старшинству часть (L) и в конечном счете — младшую часть (n).

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

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

Пользователи адресуют компьютеры иерархическими символьными именами, которые автоматически заменяются в сообщениях, передаваемых по сети, иерархическими числовыми

адресами. С помощью этих числовых адресов сообщения передаются из одной сети в другую, а после доставки сообщения в сеть назначения вместо иерархического числового адреса используется плоский аппаратный адрес компьютера. Проблема установления соответствия между адресами различных типов может решаться как централизованными, так и распределенными средствами.

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

Пример: DNS (Domain Name System) стека TCP/IP.

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

Достоинство распределенного подхода в том, что в этом случае не нужно выделять специальный компьютер, который к тому же часто требует ручного задания таблицы соответствия адресов. Недостатком его является необходимость широковещательных сообщений, перегружающих сеть. Именно поэтому распределенный подход используется в небольших сетях, а централизованный — в больших.

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

Коммутация

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

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

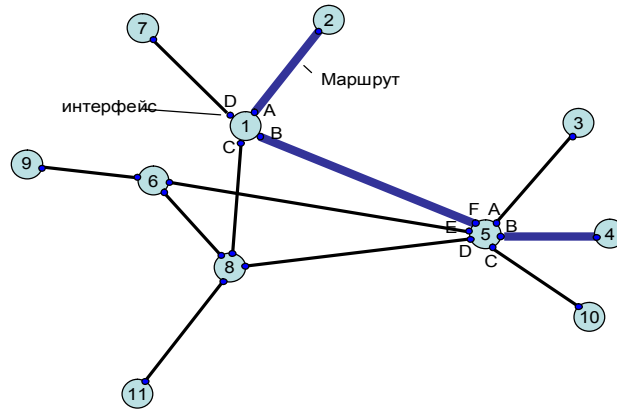


Рис. 2.12. Коммутация абонентов через сеть транзитных узлов

Например, в сети, показанной на рис. 2.12, узлы 2 и 4, непосредственно между собой не связанные и вынуждены передавать данные через транзитные узлы, в качестве которых могут выступить, например, узлы 1 и 5. Узел 1 должен выполнить передачу данных между своими интерфейсами А и В, а узел 5 — между интерфейсами F и В. В данном случае маршрутом является последовательность: 2-1-5-4, где 2 — узел-отправитель, 1 и 5 — транзитные узлы, 4 — узел-получатель.

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

1. Определение информационных потоков, для которых требуется прокладывать маршруты.
2. Маршрутизация потоков.
3. Продвижение потоков, то есть распознавание потоков и их локальная коммутация на каждом транзитном узле.
4. Мультиплексирование и демультимплексирование потоков.

Определение информационных потоков

Понятно, что через один транзитный узел может проходить несколько маршрутов, например, через узел 5 (см. рис. 2.12) проходят как минимум все данные, направляемые узлом 4 каждому из остальных узлов, а также все данные, поступающие в узлы 3, 4 и 10. Транзитный узел должен уметь *распознавать* поступающие на него потоки данных, для того чтобы обеспечивать передачу каждого из них именно на тот свой интерфейс, который ведет к нужному узлу.

Информационным потоком, или **потоком данных**, называют непрерывную последовательность данных, объединенных набором общих признаков, выделяющих их из общего сетевого трафика

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

Признаки потока могут иметь *глобальное* или *локальное* значение — в первом случае они однозначно определяют, поток в пределах всей сети, а во втором — в пределах одного транзитного узла. Пара адресов конечных узлов для идентификации потока — это пример глобального признака. Примером признака, локально определяющего поток в пределах устройства, может служить номер (идентификатор) интерфейса данного устройства, на который должны быть направлены данные.

Метка потока — это особый тип признака. Она представляет собой некоторое число, которое несут все данные потока.

Глобальная метка назначается данным потока и не меняет своего значения на всем протяжении его пути следования от узла источника до узла назначения, таким образом, она уникально определяет поток в пределах сети.

В некоторых технологиях используются **локальные метки** потока, динамически меняющие свое значение при передаче данных от одного узла к другому.

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

Маршрутизация

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

- определение маршрута;
- оповещение сети о выбранном маршруте.

Определить маршрут - это значит выбрать последовательность транзитных узлов и их интерфейсов, через которые надо передавать данные, чтобы доставить их адресату.

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

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

Продвижение данных

Для каждой пары абонентов эта операция может быть представлена несколькими (по числу транзитных узлов) **локальными** операциями коммутации. Прежде всего, отправитель должен выставить данные на тот свой интерфейс, с которого начинается найденный маршрут, а все транзитные узлы должны соответствующим образом выполнить «переброску» данных с одного своего интерфейса на другой, другими словами, выполнить **коммутацию интерфейсов**.

Устройство, функциональным назначением которого является коммутация интерфейсов, называется коммутатором (рис. 2.14).

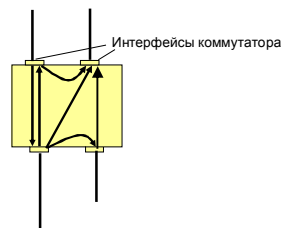


Рис. 2.14. Коммутатор

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

Коммутатором может быть как специализированное устройство, так и универсальный компьютер со встроенным программным механизмом коммутации, в этом случае **коммутатор называется программным**.

Компьютер может совмещать функции коммутации данных с выполнением своих обычных функций как конечного узла. Однако во многих случаях более рациональным является решение, в соответствии с которым **некоторые узлы в сети выделяются специально для коммутации**. Эти узлы образуют **коммутационную сеть**, к которой подключаются все остальные.

На рис. 2.15 показана коммутационная сеть, образованная из узлов 1, 5, 6 и 8, к которой подключаются конечные узлы 2, 3, 4, 7, 9 и 10.

Мультиплексирование и демultipлексирование

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

В последнем случае к задаче распознавания потоков добавляется задача **демultipлексирования**, то есть **разделения суммарного агрегированного потока на несколько составляющих его потоков**.

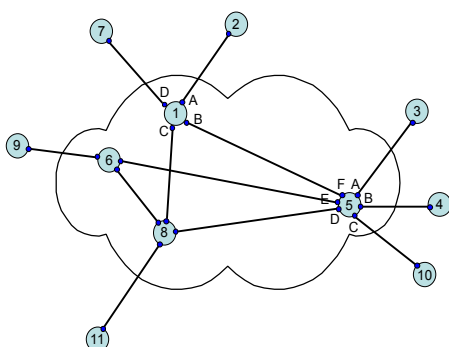


Рис. 2.15. Коммутационная сеть

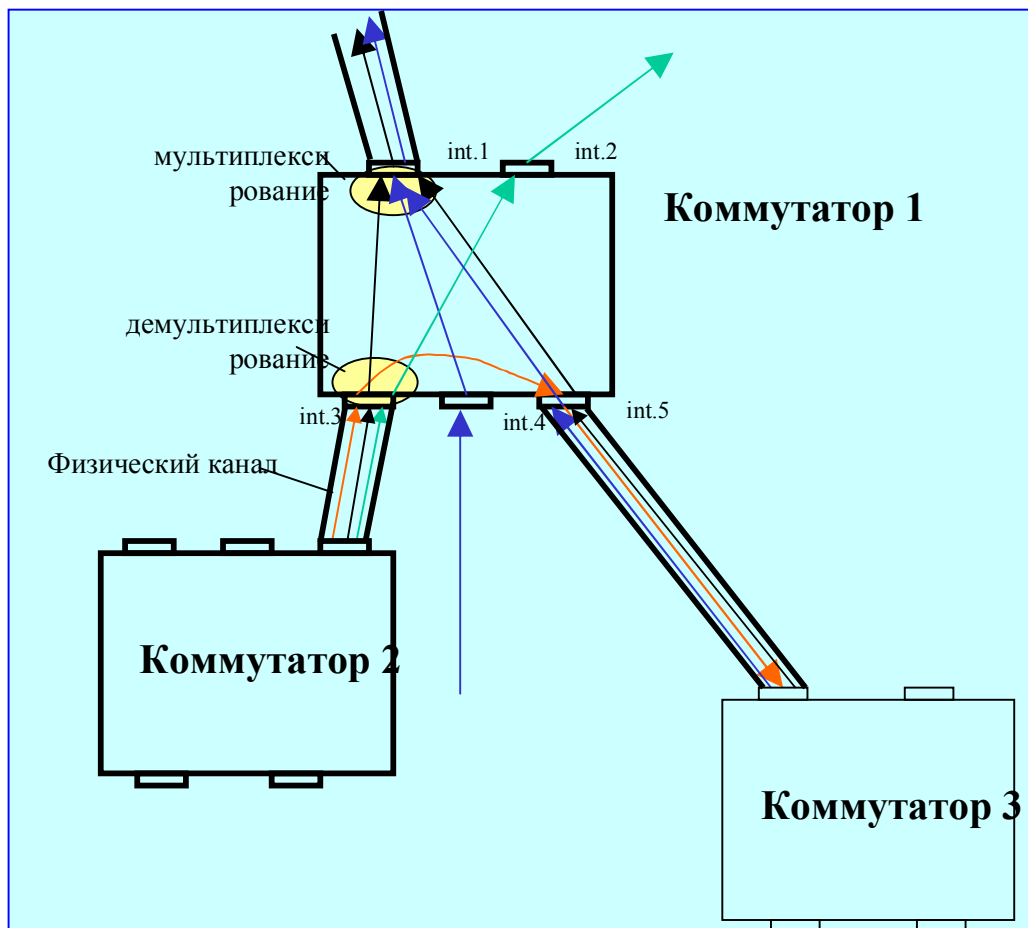


Рис. 2.16. Операции мультиплексирования и демультиплексирования потоков при коммутации

Как правило, операцию коммутации сопровождает также обратная операция — мультиплексирование. При мультиплексировании из нескольких отдельных потоков образуется общий агрегированный поток, который можно передавать по одному физическому каналу связи.

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

На рис. 2.16 показан фрагмент сети, состоящий из трех коммутаторов. Коммутатор 1 имеет пять сетевых интерфейсов. Рассмотрим, что происходит на интерфейсе Инт.1. Сюда поступают данные с трех интерфейсов — Инт.3, Инт.4 и Инт.5. Все их надо передать в общий физический канал, то есть выполнить операцию мультиплексирования. Мультиплексирование является способом разделения имеющегося одного физического канала между несколькими одновременно протекающими сеансами связи между абонентами сети.

Частный случай коммутатора, у которого все входящие информационные потоки коммутируются на один входной интерфейс, где они мультиплексируются в один агрегированный поток, называется мультиплексором (рис.2.17,а). Коммутатор, который имеет один входной интерфейс и несколько выходных, называется демультиплексором (рис. 2. 17, б).

Разделение
анием с
ый поток
і канал в
вание с

Распространено также **частотное разделение** канала, когда каждый поток передает данные в выделенном ему частотном диапазоне. Иначе этот способ мультиплексирования называется **частотным мультиплексированием (Frequency Division Multiplexing, FDM)**. Частотное мультиплексирование используется при передаче информации в аналоговом виде.

Технология мультиплексирования должна позволять получателю такого суммарного потока выполнять обратную операцию — **разделение (демультиплексирование)** данных на слагаемые потоки. На интерфейсе Инт.3 коммутатор выполняет демультиплексирование потока на три составляющих его подпотока. Один из них он передает на интерфейс Инт.1, другой — на Инт.2, а третий — на Инт.5. А вот на интерфейсе Инт.2 нет необходимости выполнять мультиплексирование или демультиплексирование — этот интерфейс выделен одному потоку в монопольное использование. Вообще говоря, на каждом интерфейсе могут одновременно выполняться обе функции — мультиплексирования и демультиплексирования.

На **рис. 2.17** показано оборудование, называемое мультиплексором и демультиплексором, являющееся частным случаем коммутатора.

Разделяемая среда передачи данных

Еще одним параметром разделяемого канала связи является **количество подключенных к нему узлов**. В приведенных выше примерах к каждому каналу связи подключались только два взаимодействующих узла, точнее - два интерфейса (рис. 2.18. а и б). В телекоммуникационных сетях используется и другой вид подключения, когда к одному каналу подключается несколько интерфейсов (рис. 2.18. в). Такое множественное подключение интерфейсов порождает уже рассматривавшуюся выше топологию «общая шина», иногда называемую также **шлейфовым подключением**. Во всех этих случаях возникает проблема организации совместного использования канала несколькими интерфейсами.

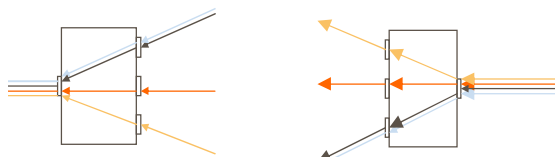


Рис. 2.17. Мультиплексор (а) и демультиплексор (б)

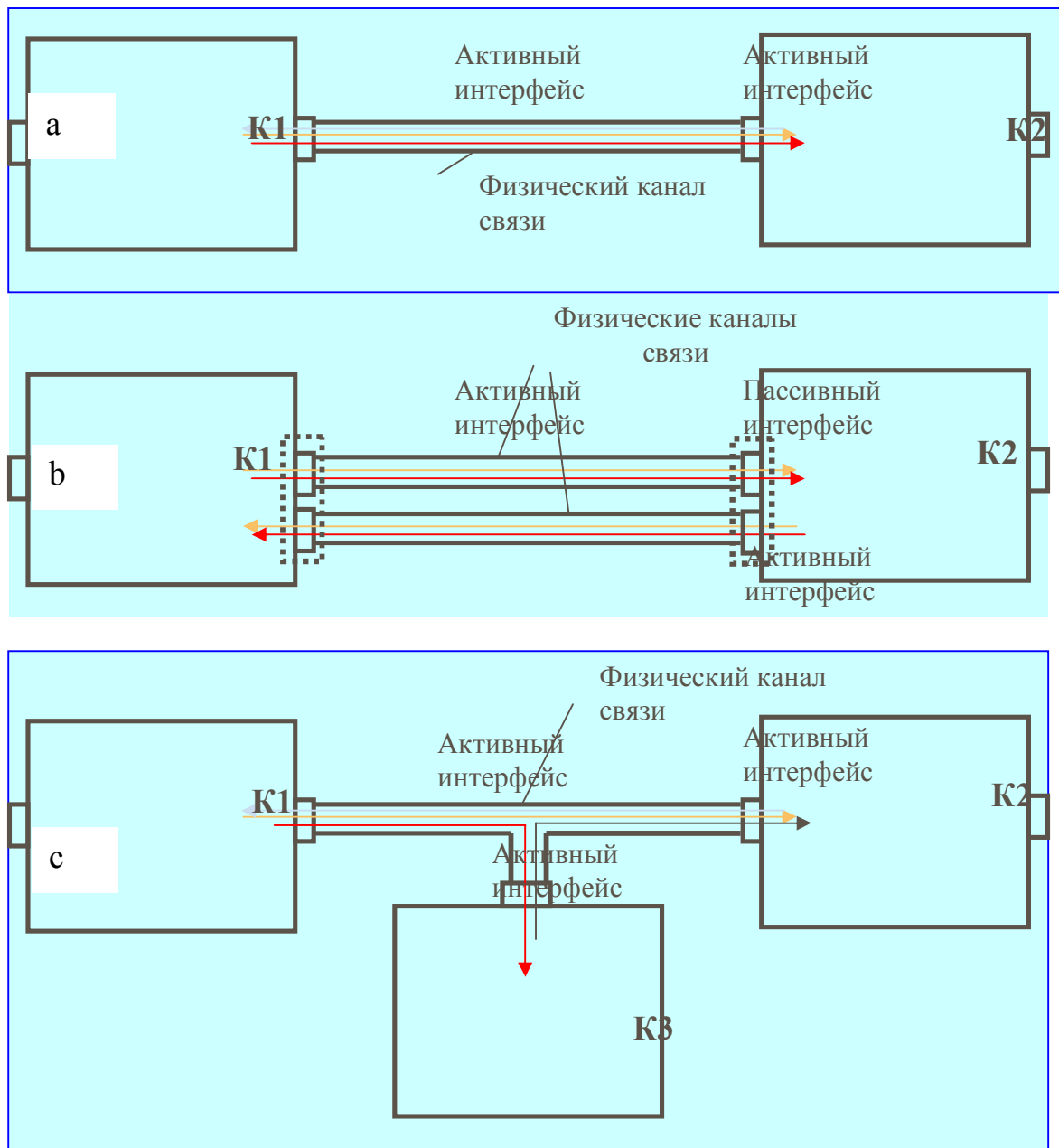


Рис. 2.18. Совместное использование канала связи

Возможны различные варианты разделения каналов связи между интерфейсами. На рис. 2.18, а коммутаторы K1 и K2 связаны двумя однонаправленными физическими каналами, то есть такими, по которым информация может передаваться только в одном направлении. В этом случае передающий интерфейс является активным, и физическая среда передачи находится под его управлением. Пассивный интерфейс только принимает данные. Проблема разделения канала между интерфейсами здесь отсутствует. (Заметим, однако, что задача мультиплексирования потоков данных в канале при этом сохраняется.) На практике два однонаправленных канала, реализующие в целом дуплексную связь между двумя устройствами, обычно рассматриваются как один дуплексный канал, а пара интерфейсов одного устройства - как передающая и принимающая части одного и того же интерфейса. На рис. 2.18. б коммутаторы K1 и K2 связаны каналом, который может передавать данные в обе стороны, но только попеременно. При этом возникает необходимость в механизме синхронизации доступа интерфейсов K1 и K2 к такому каналу. Обобщением этого варианта является случай,

показанный на рис. 2.18, с. когда к каналу связи подключаются несколько (больше двух) интерфейсов, образуя общую шину.

Совместно используемый несколькими интерфейсами физический канал называют разделяемым (shared). Часто применяют также термин **разделяемая среда передачи данных.**

Среда передачи, которая используется попеременно всеми устройствами сети, называется разделяемой средой.

Разделяемые каналы связи требуются не только для связей типа коммутатор-коммутатор, но и для связей компьютер-коммутатор и компьютер-компьютер.

Нужно подчеркнуть, что термин «разделяемая среда» традиционно относят именно к случаю **разделения канала между интерфейсами** и практически никогда — к случаю разделения канала между потоками.

Существуют различные способы решения задачи организации совместного доступа к разделяемым линиям связи. Одни из них подразумевают централизованный подход, когда доступом управляет специальное устройство — **арбитр**, другие — децентрализованный. Внутри компьютера проблемы разделения линий связи между различными модулями также существуют — примером является доступ к системной шине, которым управляет либо процессор, либо специальный арбитр шины. В сетях организация совместного доступа к линиям связи имеет свою специфику из-за существенно большего времени распространения сигналов по линиям связи, поэтому процедуры согласования доступа к линии связи могут занимать слишком большой промежуток времени и приводить к значительным потерям производительности сети.

Именно по этой причине **разделяемые между интерфейсами среды практически не используются в глобальных сетях.**

В локальных же сетях разделяемые среды используются достаточно часто благодаря простоте и экономичности их реализации. Этот подход, в частности, применяется в доминирующей сегодня в локальных сетях технологии Ethernet, а также в популярных в прошлом технологиях Token Ring и FDDI.

Однако в последние годы стала преобладать другая тенденция — отказ от разделяемых сред передачи данных и в локальных сетях. Это связано с тем, что за достигаемое таким образом удешевление сети приходится расплачиваться производительностью.

Сеть с разделяемой средой при большом количестве узлов будет работать всегда медленнее, чем аналогичная сеть с индивидуальными двухточечными линиями связями, так как **пропускная способность линии связи при ее совместном использовании делится между несколькими компьютерами сети.**

И, тем не менее не только в классических, но и в некоторых совсем новых технологиях, разработанных для локальных сетей, сохраняется режим разделяемых линий связи. Например, разработчики технологии Gigabit Ethernet, принятой в 1998 году в качестве нового стандарта, включили режим разделения среды в свои спецификации наряду с режимом работы по индивидуальным линиям связи.

Типы коммутации

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

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

- разделение среды передачи.

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

Сети с коммутацией каналов имеют более богатую историю, они происходят от первых телефонных сетей. Сети с коммутацией пакетов сравнительно молоды, они появились в конце 60-х годов как результат экспериментов с первыми глобальными компьютерными сетями. Каждая из этих схем имеет свои достоинства и недостатки, но по долгосрочным прогнозам многих специалистов будущее принадлежит технологии коммутации пакетов, как более гибкой и универсальной.

Выводы

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

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

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

При связывании в сеть более двух компьютеров возникают проблемы:

- выбора топологии (полносвязной, ячеистой, звезды, кольца, общей шины, иерархического дерева, произвольной);
- способа адресации (плоского или иерархического, числового или символического);
- способа разделения линий связи;
- механизма коммутации.

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

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

Вопросы и задания

1. Какая информация передается по каналу, связывающему внешние интерфейсы компьютера и периферийного устройства?
2. Какие компоненты включает интерфейс устройства?

3. Какие задачи решает ОС при обмене с периферийным устройством?
4. Какие функции возлагаются на драйвер периферийного устройства?
5. Дайте определение понятия «топология».
6. К какому типу топологии можно отнести структуру, образованную тремя связанными друг с другом узлами (в виде треугольника)?
7. К какому типу топологии можно отнести структуру, образованную четырьмя связанными друг с другом узлами (в виде квадрата)?
8. К какому типу топологии можно отнести структуру, образованную тремя последовательно соединенными друг с другом узлами (последний не связан с первым)?
9. Частным случаем какой топологии является общая шина:
 - полносвязная;
 - кольцо;
 - звезда.
10. Какая из известных топологий обладает повышенной надежностью?
11. Какой тип топологии наиболее распространен сегодня в локальных сетях?
12. Какие требования предъявляются к системе адресации?
13. К какому типу можно отнести следующие адреса:
 - www.oiiifer.net;
 - 20-34-a2-00-c2-27;
 - 128.145.23.170.
14. Чем неравномерный поток данных отличается от равномерного?
15. Какие параметры передаваемых данных могут служить признаком потока?
16. Какие из утверждений о маршруте, на ваш взгляд, не всегда верны:
 - маршрут - это последовательность промежуточных узлов (интерфейсов), которые проходят данные по пути от отправителя к получателю;
 - при определении маршрута всегда выбирается один из нескольких возможных путей;
 - каждый маршрут назначается для определенного потока данных;
 - из нескольких возможных маршрутов всегда выбирается оптимальный.
17. Опишите основные подходы и критерии, используемые при выборе маршрута.
18. Какие из этих утверждений могут быть в некоторых случаях верными:
 - маршруты фиксируются в коммутаторах путем жесткого соединения пар интерфейсов;
 - маршруты определяются администратором и заносятся вручную в специальную таблицу;
 - таблица маршрутов строится автоматически сетевым программно-аппаратным обеспечением;
 - для каждого коммутатора строится своя таблица маршрутов, которая на нем и хранится.
19. Какое из этих устройств можно назвать коммутатором:
 - электрический выключатель;
 - автоматическая телефонная станция;
 - маршрутизатор;
 - мост;
 - мультиплексор;
 - ни одно из названных.
20. Какие методы используются при мультиплексировании?
21. Объясните различия между разделением среды передачи и мультиплексированием.
22. Опишите, какие основные задачи нужно решить, чтобы обеспечить информационное взаимодействие любой пары абонентов в коммуникационной сети любого типа.
23. Как представление общего городского трафика в виде нескольких различных потоков позволяет рационализировать управление городским транспортом?

24. Пусть в сети существует несколько маршрутов между двумя конечными узлами А и В. Перечислите достоинства и недостатки следующих вариантов передачи данных между этими узлами:

- использовать все имеющиеся маршруты для параллельной передачи данных;
- передавать все данные по одному оптимальному по некоторому критерию маршруту;
- использовать несколько маршрутов из набора всех возможных маршрутов и разделять между ними передаваемые данные.

Какое правило можно применить для определения маршрута передачи очередного пакета в последнем из перечисленных случаев?

ГЛАВА 3. Архитектура и стандартизация сетей

Архитектура подразумевает представление сети в виде системы элементов, каждый из которых выполняет определенную частную функцию, при этом все элементы вместе согласованно решают общую задачу взаимодействия компьютеров. Другими словами, архитектура сети отражает декомпозицию общей задачи взаимодействия компьютеров на отдельные подзадачи, которые должны решаться отдельными элементами сети.

Одним из важных элементов архитектуры сети является коммуникационный протокол — формализованный набор правил взаимодействия узлов сети.

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

Стандартная архитектура компьютерной сети определяет также распределение протоколов между элементами сети — конечными узлами (компьютерами) и промежуточными узлами (повторителями, концентраторами, мостами, коммутаторами, маршрутизаторами, шлюзами).

Промежуточные узлы выполняют только транспортные функции стека протоколов, передавая трафик между конечными узлами.

Конечные узлы поддерживают весь стек протоколов, предоставляя информационные услуги, например веб-сервис. Такое распределение функций означает смещение «интеллекта» сети на ее периферию.

Декомпозиция задачи сетевого взаимодействия

Организация взаимодействия между устройствами сети является сложной задачей. Для решения сложных задач используется известный универсальный прием — декомпозиция, то есть разбиение одной сложной задачи на несколько более простых задач-модулей. Декомпозиция состоит в четком определении:

- функций каждого модуля,
- порядке их взаимодействия (то есть межмодульных интерфейсов).

В результате такого логического упрощения задачи появляется возможность независимого тестирования, разработки и модификации модулей. Так, любой из показанных на рис. 3.1 модулей может быть переписан заново.

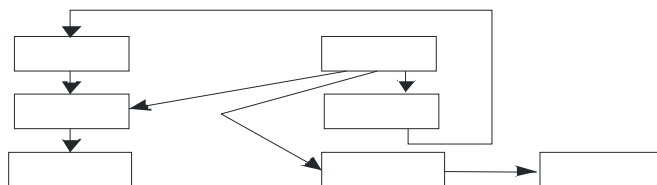


Рис. 3.1. Пример декомпозиции задачи

Многоуровневый подход

Еще более эффективной концепцией, развивающей идею декомпозиции, является **многоуровневый подход**. После представления исходной задачи в виде множества модулей **эти модули группируют и упорядочивают по уровням**, образуя иерархию. В соответствии с принципом иерархии для каждого промежуточного уровня можно указать непосредственно примыкающие к нему соседние вышележащий и нижележащий уровни (рис. 3.2).

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

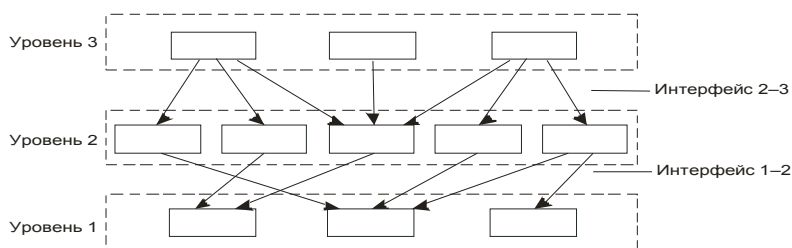


Рис. 3.2. Многоуровневый подход — создание иерархии задач

Межуровневый интерфейс, называемый также интерфейсом услуг, определяет набор функций, которые нижележащий уровень предоставляет вышележащему (рис. 3.2).

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

Протокол и стек протоколов

Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе обмена сообщениями участвуют, по меньшей мере, **две стороны**, то есть в данном случае **необходимо организовать согласованную работу двух иерархий аппаратных и программных средств, работающих на разных компьютерах**. Оба участника сетевого обмена должны принять множество соглашений. Например, они должны согласовать уровни и форму электрических сигналов, способ определения размера сообщений, договориться о методах контроля достоверности и т. п. Другими словами, соглашения должны быть приняты на всех уровнях, начиная от самого низкого — уровня передачи битов и заканчивая самым высоким, реализующим обслуживание пользователей сети. На рис. 3.5 показана модель взаимодействия двух узлов.

С каждой стороны средства взаимодействия представлены четырьмя уровнями. **Каждый уровень поддерживает интерфейсы двух типов:**

1. Интерфейсы услуг с выше- и нижележащим уровнями «своей» иерархии средств.
2. Интерфейс со средствами взаимодействия другой стороны, расположенными на том же уровне иерархии. Этот тип интерфейса называют **протоколом**.

Таким образом, **протокол всегда является одноранговым интерфейсом**.

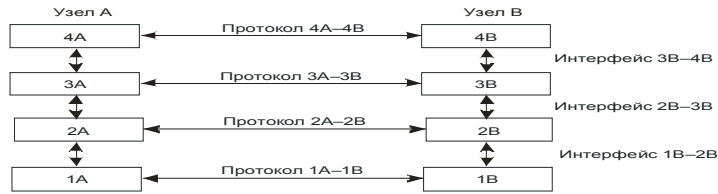


Рис. 3.5. Взаимодействие двух узлов

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

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

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

Программный модуль, реализующий некоторый протокол, называют протокольной сущностью, или, для краткости, тоже протоколом.

Протокольные сущности одного уровня двух взаимодействующих сторон обмениваются **сообщениями** в соответствии с определенным для них протоколом.

Сообщения состоят из заголовка и поля данных (иногда оно может отсутствовать).

Обмен сообщениями является своеобразным языком общения, с помощью которого каждая из сторон «объясняет» другой стороне, что необходимо сделать на каждом этапе взаимодействия. Работа каждого **протокольного модуля состоит в чтении, анализе заголовков поступающих к нему сообщений и выполнении связанных с этим действий**. Заголовки сообщений разных протоколов имеют разную структуру, что соответствует различиям в их функциональности. Понятно, что чем сложнее структура заголовка сообщения, тем более сложные функции возложены на соответствующий протокол.

Общая характеристика модели OSI

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

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

Назначение модели OSI состоит в обобщенном представлении средств сетевого взаимодействия. Она разрабатывалась в качестве своего рода **универсального языка сетевых специалистов, именно поэтому ее называют справочной моделью**.

Модель OSI определяет:

- уровни взаимодействия систем в сетях с коммутацией пакетов,
- стандартные названия уровней,
- функции, которые должен выполнять каждый уровень

Модель OSI не содержит описаний реализаций конкретного набора протоколов

В модели OSI (рис. 3.6) 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 программными и аппаратными средствами. Модель не включает средства взаимодействия приложений конечных пользователей. Важно различать уровень взаимодействия приложений и прикладной уровень семиуровневой модели.

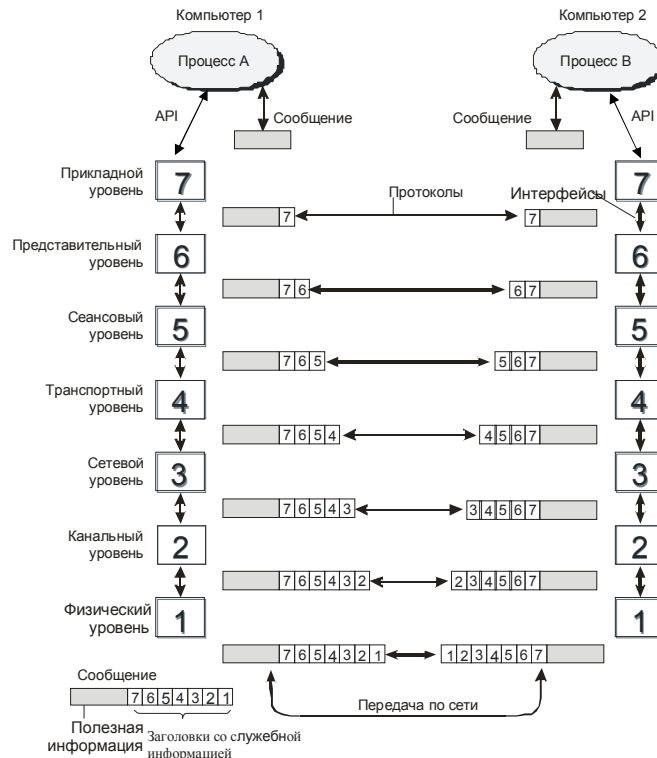


Рис. 3.6. Модель взаимодействия открытых систем ISO/OSI

Приложения могут реализовывать собственные протоколы взаимодействия, используя для этих целей многоуровневую совокупность системных средств. Именно для этого в распоряжение программистов предоставляется **прикладной программный интерфейс (Application Program Interface, API)**.

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 прикладного уровня, выполняет требуемые действия и добавляет к сообщению собственную служебную информацию — заголовок уровня представления, в котором **содержатся указания для протокола уровня представления машины-адресата**. Полученное в результате сообщение передается вниз сеансовому уровню, который в свою очередь добавляет свой заголовок и т. д. (Некоторые реализации протоколов помещают служебную информацию не только в начале сообщения в виде заголовка, но и в конце в виде так

называемого **концевика**.) Наконец, сообщение достигает нижнего, физического уровня, который, собственно, и передает его по линиям связи машине-адресату. К этому моменту сообщение «обрастает» заголовками всех уровней (рис. 3.7).

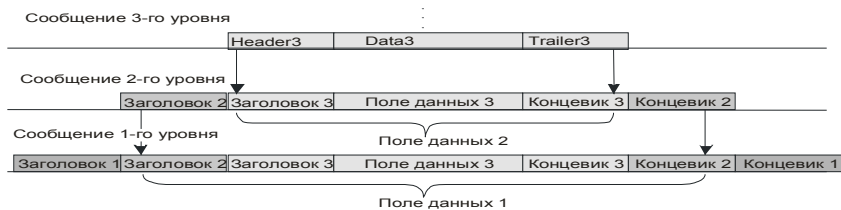


Рис. 3.7. Вложенность сообщений различных уровней

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

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

Физический уровень

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

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

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

Физический уровень не вникает в смысл информации, которую он перелает. 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 PDU (протокольный блок данных) обычно носит название **кадр (frame)**.

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

- **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)**. Примерами технологий локальных сетей, применение которых ограничено типовыми топологиями, являются Ethernet, FDDI, Token Ring.
- **В глобальных сетях** канальный уровень должен обеспечивать доставку кадра только между *двумя соседними узлами*, соединенными индивидуальной линией связи. Примерами двухточечных протоколов (как часто называют такие протоколы) могут служить широко распространенные протоколы PPP (Point to Point Protocol) и HDLC. На основе двухточечных связей могут быть построены сети произвольной топологии.

Для связи локальных сетей между собой или для доставки сообщений между **любыми конечными узлами глобальной сети** используются средства более высокого сетевого уровня.

Одной из функций канального уровня является **поддержание интерфейсов с нижележащим физическим уровнем и вышележащим сетевым уровнем**. Сетевой уровень направляет канальному уровню пакет для передачи в сеть или принимает от него пакет, полученный из сети. **Физический уровень используется канальным** как инструмент, который принимает и передает в сеть последовательности битов.

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

Одной из задач канального уровня является **обнаружение и коррекции ошибок**. Для этого канальный уровень фиксирует границы кадра, помещая специальную последовательность битов в его начало и конец (**преамбула** (7 байт 10101010, 8-ой байт 10101011), **флаг** (01111110), необходимые для синхронизации кадра), а затем вставляет в кадр **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 или нет искажены данные кадра в процессе передачи по сети.

Однако прежде, чем переправить кадр физическому уровню для непосредственной передачи данных в сеть, канальному уровню может потребоваться решить еще одну важную задачу. Если в сети используется разделяемая среда, то прежде чем физический уровень начнет передавать данные, канальный уровень **должен проверить доступность среды**.

Если разделяемая среда освободилась (когда она не используется, то такая проверка, конечно, пропускается), кадр передается средствами физического уровня в сеть, проходит по каналу связи и поступает в виде последовательности битов в распоряжение физического уровня узла назначения. Этот уровень в свою очередь передает полученные биты «наверх» канальному уровню своего узла. Последний группирует биты в кадры, снова вычисляет контрольную последовательность кадра полученных данных по тому же алгоритму и сравнивает результат с контрольной последовательностью, переданной в кадре. Если они совпадают, кадр считается правильным. Если же контрольные последовательности не совпадают, фиксируется ошибка. В функции канального уровня входит не только обнаружение ошибок, но и **исправление их за счет повторной передачи поврежденных кадров**. Однако эта функция не является обязательной и в некоторых реализациях канального уровня она отсутствует, например в Ethernet, Token Ring, FDDI и Frame Relay.

Все вышеназванные функции канального уровня иногда выделяют в отдельный **подуровень управления доступом к среде (Media Access Control, MAC)**, а функции интерфейса с вышележащим сетевым уровнем выделяют в **подуровень управления логической связью (Logical Link Control)**

Протоколы канального уровня реализуются компьютерами, мостами, коммутаторами, маршрутизаторами, шлюзами. В компьютерах функции канального уровня реализуются совместными усилиями сетевых адаптеров и их драйверов.

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

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

Это утверждение в еще большей степени справедливо для **глобальных сетей**, в которых протокол канального уровня реализует достаточно простую функцию **передачи данных между соседними узлами**.

В качестве примера приведены на **рис. 3.8** приведены протоколы сетевых технологий семейств Ethernet и WLAN, а также технологии Token Ring, разработанных 802 комитетом Института инженеров по электронике и радиоэлектронике (IEEE).

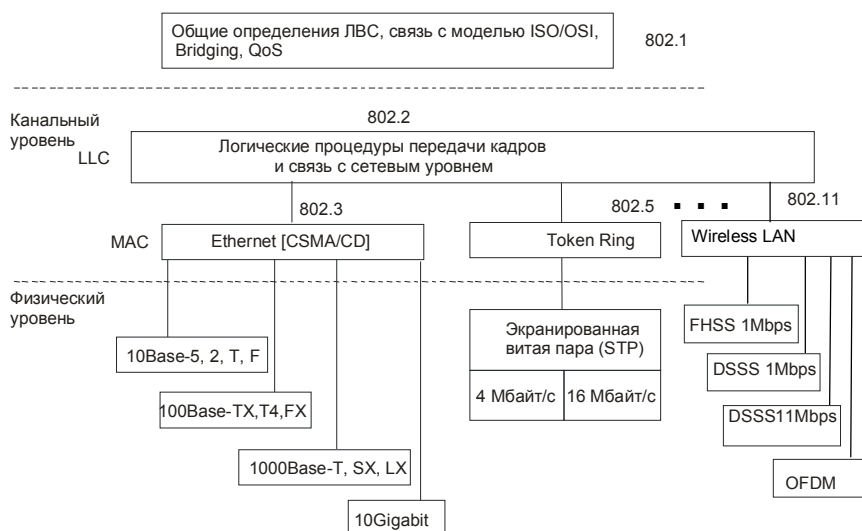


Рис. 3.8. Структура стандартов IEEE 802.x

Сетевой уровень

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

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

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

На **рис. 3.8a** показаны несколько сетей, каждая из которых использует собственную технологию канального уровня: Ethernet, FDDI, Token Ring, ATM, Frame Relay. **На базе этих технологий каждая из указанных сетей может связывать между собой любых пользователей, но только своей сети, и не способна обеспечить передачу данных в другую сеть.** Причина такого положения вещей очевидна и кроется в существенных отличиях одной технологии от другой. Даже наиболее близкие технологии LAN — Ethernet, FDDI, Token Ring, — имеющие одну и ту же систему адресации (адреса подуровня MAC, называемые **MAC-адресами**), отличаются друг от друга форматом используемых кадров и логикой работы протоколов. Еще больше отличий между технологиями LAN и WAN. Во многих технологиях WAN задействована техника предварительно устанавливаемых виртуальных каналов, идентификаторы которых применяются в качестве адресов. Все технологии имеют собственные форматы кадров (в технологии ATM кадр даже называется иначе — ячейкой) и, конечно, собственные стеки протоколов.

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

Функции сетевого уровня реализуются:

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

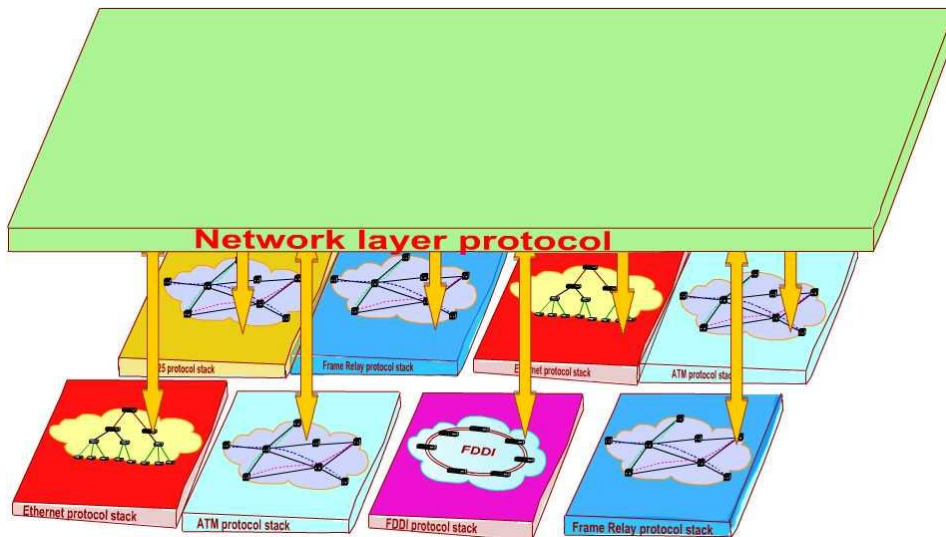


Рис. 3.8а. Необходимость сетевого уровня

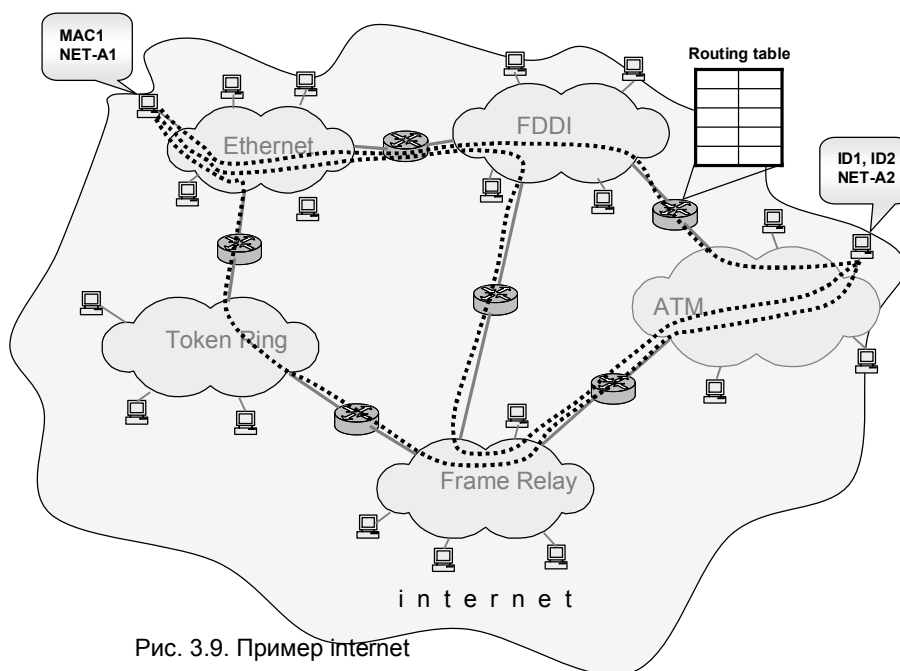


Рис. 3.9. Пример Internet

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.

Чтобы связать сети, показанные на рис. 3.8а. необходимо соединить все эти сети маршрутизаторами и установить протокольные модули сетевого уровня на все конечные узлы пользователей, которые хотели бы связываться через составную сеть (рис. 3.9).

Данные, которые необходимо передать через составную сеть, поступают на сетевой уровень от вышележащего транспортного уровня. Эти данные снабжаются заголовком сетевого

уровня. Данные вместе с заголовком образуют **пакет** — так называется **PDU сетевого уровня**. Заголовок пакета сетевого уровня имеет унифицированный формат, не зависящий от форматов кадров канального уровня тех сетей, которые могут входить в составную сеть, и несет наряду с другой служебной информацией данные об адресе назначения этого пакета.

Для того чтобы **протоколы сетевого уровня могли доставлять пакеты любому узлу составной сети, эти узлы должны иметь адреса, уникальные в пределах данной составной сети**. Такие адреса называются **сетевыми, или глобальными**. Каждый узел составной сети, который намерен обмениваться данными с другими узлами составной сети, должен иметь **сетевой адрес наряду с адресом, назначенным ему на канальном уровне**. Например, на рис. 3.9 компьютер в сети Ethernet, входящей в составную сеть, имеет адрес канального уровня MAC1 и адрес сетевого уровня NET-A1; аналогично в сети ATM узел, адресуемый идентификаторами виртуальных каналов ID1 и ID2, имеет сетевой адрес NET-A2. В пакете в качестве адреса назначения должен быть указан адрес сетевого уровня, на основании которого определяется маршрут пакета. **Определение маршрута** является важной задачей сетевого уровня. **Маршрут описывается последовательностью сетей (или маршрутизаторов), через которые должен пройти пакет, чтобы попасть к адресату**. Например, на рис. 3.9 штриховой линией показано 3 маршрута, по которым могут быть переданы данные от компьютера А к компьютеру В. Маршрутизатор собирает информацию о топологии связей между сетями и на ее основании строит **таблицы коммутации**, которые в данном случае носят специальное название **таблиц маршрутизации**.

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

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

На сетевом уровне определяются **два вида протоколов**. Первый вид — **маршрутизируемые протоколы** — реализуют продвижение пакетов через сеть. Именно эти протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня. Пример: **межсетевой протокол (Internet Protocol, IP)**. В его задачу входит продвижение пакета между сетями — от одного маршрутизатора до другого до тех пор, пока пакет не попадет и сеть назначения.

Однако часто к сетевому уровню относят и **другой вид протоколов, называемых маршрутизирующими протоколами, или протоколами маршрутизации**. С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений, на основании которой осуществляется выбор маршрута продвижения пакетов. Пример: **протоколы маршрутизации RIP, OSPP, BGP**.

Транспортный уровень

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

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

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

Выбор класса сервиса транспортного уровня определяется:

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

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

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

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

Сеансовый уровень

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

Уровень представления

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

Прикладной уровень

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

Существует очень большое разнообразие протоколов и соответствующих служб прикладного уровня. 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.**

Модель OSI и сети с коммутацией каналов

Как уже было упомянуто, **модель OSI** описывает процесс взаимодействия устройств в сети с **коммутацией пакетов**.

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

Для сетей с коммутацией пакетов сети с коммутацией каналов предоставляют сервис физического уровня, несмотря на то, что сами устроены достаточно сложно и поддерживают собственную иерархию протоколов.

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

используют поверх этого физического канала какой-либо двухточечный протокол канального уровня.

Сетевая технология Ethernet.

Стандарт Ethernet был принят в 1984 году. Основной принцип, положенный в основу Ethernet, - случайный метод доступа к разделяемой среде передачи данных. В качестве такой среды может использоваться толстый или тонкий коаксиальный кабель, витая пара, оптоволокно или радиоволны (кстати, первой сетью, построенной на принципе случайного доступа к разделяемой среде, была радиосеть Aloha Гавайского университета).

Ethernet - Базовая технология локальной вычислительной сети, использующая метод множественного доступа к среде передачи с прослушиванием несущей и обнаружением коллизий (CSMA/CD – Carrier Sense Multiple-Access/Collision Detection). Топология – общая шина (рис. 3.10). Скорость передачи 10 Мб/с. Протокол стандартизован в спецификации IEEE 802.3. Основа стандарта разработка компаний DEC, Intel, Xerox, 1980 г.

Стандарт разработан институтом инженеров по электронике и радиоэлектронике – (Institute of Electrical and Electronics Engineers, IEEE) – национальной организацией США, определяющей сетевые стандарты локальных вычислительных сетей. В 1981 году рабочая группа 802 этого института сформулировала основные требования, которым должны удовлетворять локальные вычислительные сети. Группа 802 определила множество стандартов, из них самыми известными являются стандарты 802.1, 802.2, 802.3, 802.5, 802.11 которые описывают общие понятия, используемые в области локальных сетей, а также стандарты на два нижних уровня сетей Ethernet, Token Ring, WLAN.

Несколько слов о стандарте 802.11. Это стандарт беспроводных сетей (WLAN – Wireless/Wale Local Area Network), в котором используется метод множественного доступа с прослушиванием несущей и исключением коллизий (CSMA/CD – Carrier Sense Multiple Access with Collision Avoidance). Этот метод требует от станции отправителя положительного подтверждения (ACK) каждого принятого кадра.

В стандарте Ethernet строго зафиксирована топология электрических связей. Компьютеры подключаются к разделяемой среде в соответствии с типовой структурой «общая шина» (рис. 3.10). С помощью разделяемой во времени шины любые два компьютера могут обмениваться данными. Управление доступом к линии связи осуществляется специальными контроллерами – сетевыми адаптерами Ethernet. Каждый компьютер, а более точно, каждый сетевой адаптер, имеет уникальный адрес. Передача данных происходит со скоростью 10 Мбит/с. Эта величина является пропускной способностью сети Ethernet.

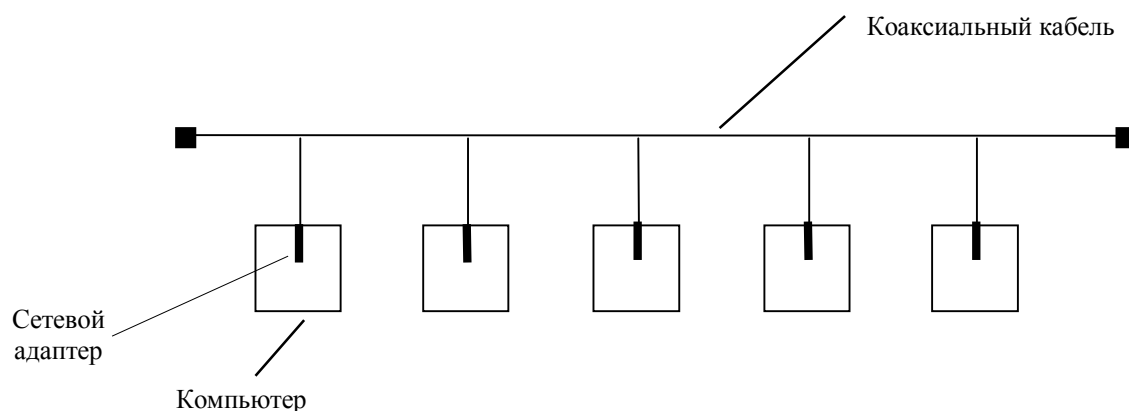


Рис. 3.10. Сеть Ethernet

Суть **случайного метода** доступа состоит в следующем. Компьютер в сети Ethernet может передавать данные по сети, только если сеть свободна, то есть если никакой другой компьютер в данный момент не занимается обменом. После того как компьютер убедился, что сеть свободна, он начинает передачу, при этом «захватывает» среду. Время монопольного использования разделяемой среды одним узлом ограничивается временем передачи одного кадра.

Кадр – это единица данных, которыми обмениваются компьютеры в локальных сетях. Кадр имеет фиксированный формат и наряду с полем данных содержит различную служебную информацию, например, адрес получателя и адрес отправителя.

Максимальная длина кадра технологии Ethernet 1518 байт, минимальная длина кадра 64 байт.

Кадр данных всегда сопровождается **преамбулой** (preamble), которая состоит из 7 байт, имеющих значения 10101010, и 8-го байта, равного 10101011. Преамбула нужна для определения начала и конца кадра (синхронизации кадра).

Сеть Ethernet устроена так, что при попадании кадра в разделяемую среду передачи данных все сетевые адаптеры одновременно начинают принимать этот кадр. Все они анализируют адрес назначения, располагающийся в одном из начальных полей кадра, и, если этот адрес совпадает с их собственным адресом, кадр помещается во внутренний буфер сетевого адаптера. Таким образом компьютер-адресат получает предназначенные ему данные.

Иногда может возникать ситуация, когда одновременно два и более компьютера решают, что сеть свободна, и начинают передавать информацию. Такая ситуация, называемая коллизией, препятствует правильной передаче данных по сети. В стандарте Ethernet предусмотрен алгоритм обнаружения и корректной обработки коллизий. Вероятность возникновения коллизии зависит от интенсивности сетевого трафика.

После обнаружения коллизии сетевые адаптеры, **которые пытались передать свои кадры, прекращают передачу** и после паузы случайной длительности пытаются снова получить доступ к среде и передать тот кадр, который вызвал коллизию.

Главным достоинством сетей Ethernet, благодаря которому они стали такими популярными, является их экономичность. Кроме того, в сетях Ethernet реализованы достаточно простые алгоритмы доступа к среде, адресации и передачи данных. Простота логики работы сети ведет к упрощению и, соответственно, удешевлению сетевых адаптеров и их драйверов. По той же причине адаптеры сети Ethernet обладают высокой надежностью.

Еще одним свойством сетей Ethernet является их хорошая расширяемость, то есть легкость подключения новых узлов.

Стек TCP/IP

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

Поскольку стек TCP/IP изначально создавался для Интернета, он имеет много особенностей, дающих ему преимущество перед другими протоколами, когда речь заходит **о построении сетей, включающих глобальные связи**. В частности, очень полезным свойством, делающим возможным применение этого протокола в больших сетях, является его **способность фрагментировать пакеты**. Действительно, большая составная часть часто состоит из сетей, построенных на совершенно разных принципах. В каждой из этих сетей может быть собственная величина максимальной длины единицы передаваемых данных (**кадра**). В таком случае при переходе из одной сети, имеющей большую максимальную длину, в сеть с меньшей максимальной длиной может возникнуть необходимость деления передаваемого кадра на несколько частей. **Протокол IP стека TCP/IP эффективно решает эту задачу.**

Другой особенностью технологии TCP/IP является *гибкая система адресации*, позволяющая проще, чем другие протоколы аналогичного назначения включать в составную сеть сети разных технологий. Это свойство также способствует применению стека TCP/IP для построения больших **гетерогенных сетей**.

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

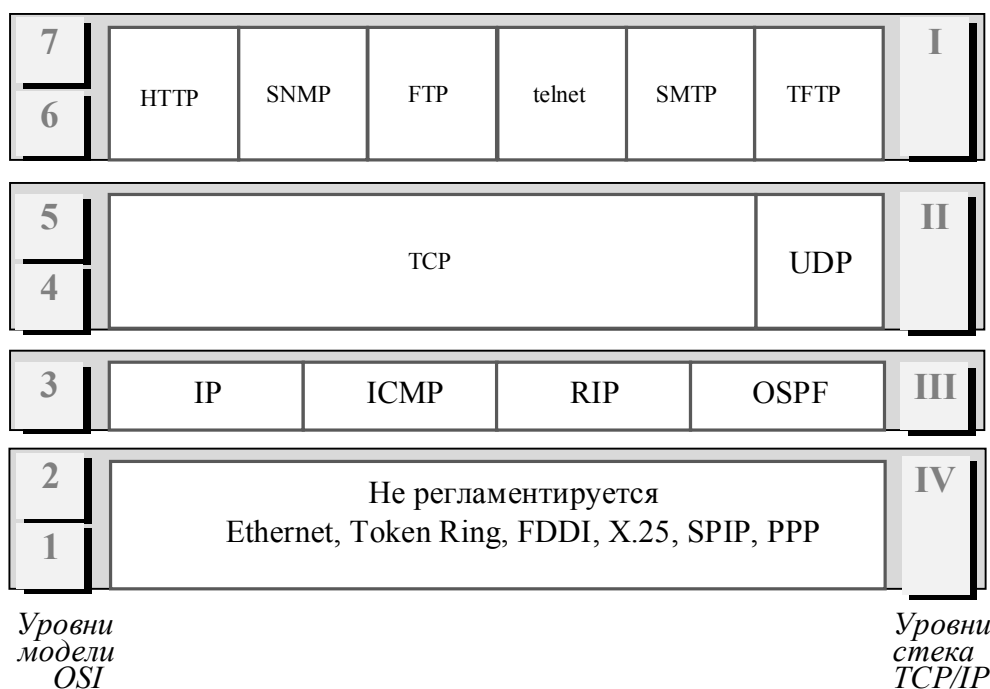
Однако, как и всегда, за получаемые преимущества надо платить, и платой здесь оказываются *высокие требования к ресурсам и сложность администрирования* IP-сетей. **Мощные функциональные возможности протоколов стека TCP/IP требуют для своей реализации больших вычислительных затрат**. Гибкая система адресации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

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

На **рис.3.11** приведена структура стека TCP/IP. Так как стек TCP/IP был разработан до появления модели ISO/OSI, то, хотя он также имеет многоуровневую структуру, **соответствие уровней стека TCP/IP уровням модели OSI достаточно условно**.

В стеке TCP/IP определены 4 уровня.

Прикладной уровень стека TCP/IP соответствует трем верхним уровням модели OSI: **прикладному, представления и сеансовому**. Он объединяет службы, предоставляемые системой пользовательским приложениям. За долгие годы использования в сетях различных стран и организаций стек TCP/IP накопил большое количество протоколов и служб прикладного уровня. К ним относятся такие распространенные протоколы, как **протокол передачи файлов (File Transfer Protocol, FTP)**, протокол эмуляции терминала (**telnet**), **простой протокол передачи электронной почты (Simple Mail Transfer Protocol, SMTP)**, **протокол передачи гипертекста (HyperText Transfer Protocol, HTTP)** и многие другие. Протоколы прикладного уровня развертываются на хостах. В Интернете конечный узел традиционно называют *хостом*, а маршрутизатор — *шлюзом*. В этой главе мы будем также придерживаться такой терминологии.



Транспортный уровень стека TCP/IP может предоставлять вышележащему уровню два типа сервиса:

1. Гарантированную доставку обеспечивает **протокол управления передачей (Transmission Control Protocol, TCP)**;
2. Доставку по возможности, или с максимальными усилиями, обеспечивает **протокол пользовательских дейтаграмм (User Datagram Protocol, UDP)**.

Для того чтобы обеспечить надежную доставку данных протокол TCP предусматривает **установление логического соединения**, что позволяет ему нумеровать пакеты, подтверждать их прием квитанциями, в случае потери организовывать повторные передачи, распознавать и уничтожать дубликаты, доставлять прикладному уровню пакеты в том порядке, в котором они были отправлены. Этот протокол позволяет объектам на компьютере отправителе и компьютере-получателе поддерживать обмен данными в **дуплексном режиме**. TCP дает возможность **без ошибок доставить сформированный на одном из компьютеров поток байтов в любой другой компьютер, входящий в составную сеть**, TCP делит поток байтов на **фрагменты и передает их нижележащему уровню межсетевого взаимодействия**. После того как эти фрагменты будут доставлены средствами уровня межсетевого взаимодействия в пункт назначения, протокол TCP снова **соберет их в непрерывный поток байтов**.

Второй протокол этого уровня — **UDP** — является **простейшим дейтаграммным протоколом**, который используется в том случае, когда **задача надежного обмена данными либо вообще не ставится, либо решается средствами более высокого уровня** — прикладным уровнем или пользовательскими приложениями.

В функции протоколов транспортного уровня TCP и UDP входит также **исполнение роли связующего звена между прилегающими к ним прикладным уровнем и уровнем межсетевого взаимодействия**. От прикладного протокола транспортный уровень принимает задание на передачу данных с тем или иным качеством, а после выполнения рапортует ему об этом. Нижележащий уровень межсетевого взаимодействия протоколы TCP и UDP рассматривают как своего рода инструмент, не очень надежный, но способный перемещать пакет в свободном и рискованном путешествии по составной сети.

Программные модули, реализующие протоколы TCP и UDP, подобно модулям протоколов прикладного уровня, устанавливаются на **хостах**.

Сетевой уровень, называемый также **уровнем интернета**, является стержнем всей архитектуры TCP/IP. Именно этот уровень, функции которого соответствуют сетевому уровню модели OSI, **обеспечивает перемещение пакетов в пределах составной сети, образованной объединением множества сетей**. Протоколы сетевого уровня поддерживают интерфейс с вышележащим транспортным уровнем, получая от него запросы на передачу данных по составной сети, а также с нижележащим уровнем сетевых интерфейсов, о функциях которого мы расскажем далее.

Основным протоколом сетевого уровня является **межсетевой протокол (Internet Protocol, IP)**. В его задачу входит продвижение пакета между сетями — **от одного маршрутизатора до другого до тех пор, пока пакет не попадет в сеть назначения**. В отличие от протоколов прикладного и транспортного уровней протокол IP **развертывается не только на хостах, но и на всех шлюзах**. Протокол IP — это **дейтаграммный протокол, работающий без установления соединений по принципу доставки с максимальными усилиями**.

К сетевому уровню TCP/IP часто относят протоколы, выполняющие вспомогательные функции по отношению к IP. Это, прежде всего, **протоколы маршрутизации RIP и OSPF**, занимающиеся изучением топологии сети, определением маршрутов и составлением таблиц маршрутизации, на основании которых протокол IP перемещает пакеты в нужном направлении. По этой же причине к сетевому уровню могут быть отнесены еще два протокола: **протокол межсетевых управляющих сообщений (Internet Control Message Protocol, ICMP)**, предназначенный для передачи маршрутизатором источнику информации об ошибках,

возникших при передаче пакета, и **протокол групповой адресации (Internet Group Management Protocol, IGMP)**, использующийся для направления пакета сразу по нескольким адресам.

Идеологическим отличием архитектуры стека TCP/IP от многоуровневой организации других стеков является интерпретация функций самого нижнего уровня - **уровня сетевых интерфейсов**.

Напомним, что нижние уровни модели OSI (канальный и физический) реализуют большое количество функций доступа к среде передачи, формированию кадров и согласованию уровней электрических сигналов, кодированию и синхронизации и некоторые другие. Все эти весьма конкретные функции составляют суть таких протоколов обмена данными, как **Ethernet, Token Ring, PPP, HDLC** и многих других.

У нижнего уровня стека TCP/IP задача существенно проще — **он отвечает только за организацию взаимодействия с технологиями сетей, входящих в составную сеть. TCP/IP рассматривает любую сеть, входящую в составную сеть, как средство транспортировки пакетов до следующего по пути маршрутизатора.**

Задачу обеспечения интерфейса между технологией TCP/IP и любой другой технологией промежуточной сети упрощенно можно свести:

- **к определению способа упаковки (инкапсуляции) IP-пакета в единицу передаваемых данных промежуточной сети;**
- **к определению способа преобразования сетевых адресов в адреса технологии данной промежуточной сети.**

Такой подход делает **составную сеть TCP/IP открытой для включения любой сети**, какую бы внутреннюю технологию передачи данных эта сеть не использовала. **Для каждой новой технологии должны быть разработаны собственные интерфейсные средства.** Следовательно, функции этого уровня нельзя определить раз и навсегда.

Уровень сетевых интерфейсов в стеке TCP/IP не регламентируется. Он поддерживает все популярные технологии; для локальных сетей — это **Ethernet, Token Ring, FDDI, Fast Ethernet, Gigabit Ethernet**, для глобальных сетей - протоколы двухточечных соединений **SLIP и PPP**, технологии **X.25, Frame Relay, ATM**.

Обычно при появлении новой технологии локальных или глобальных сетей она быстро включается в стек TCP/IP путем разработки соответствующего документа **RFC**, определяющего метод инкапсуляции IP-пакетов в ее кадры (например, спецификация **RFC 1577**, определяющая работу протокола IP через сети ATM. появилась в 1994 году вскоре после принятия основных стандартов ATM).

Стек TCP/IP позволяет **включать в составную сеть сети независимо от того, каким количеством уровней описывается используемая в них технология.** Так, перемещение данных в сети X.25 обеспечивают собственные протоколы физического, канального и сетевого уровней (в терминологии OSI). Тем не менее стек TCP/IP рассматривает сеть X.25 наравне с другими технологиями в качестве средства транспортировки IP-пакетов между двумя пограничными шлюзами. Уровень сетевых интерфейсов обычным образом предоставляет для этой технологии способ инкапсуляции IP-пакета в пакет X.25. а также средства преобразования сетевых IP-адресов в адреса сетевого уровня X.25. Если рассматривать такую организацию сети в строгом соответствии с моделью OSI. то налицо **явное противоречие — один сетевой протокол (IP) работает поверх другого сетевого протокола (X.25).** Однако для стека TCP/IP это нормальное явление.

Каждый коммуникационный протокол оперирует некоторой единицей передаваемых данных. Названия этих единиц иногда закрепляются стандартом, а чаще просто определяются традицией. **В стеке TCP/IP за многие годы его существования образовалась устоявшаяся терминология в этой области (рис. 3.12).**

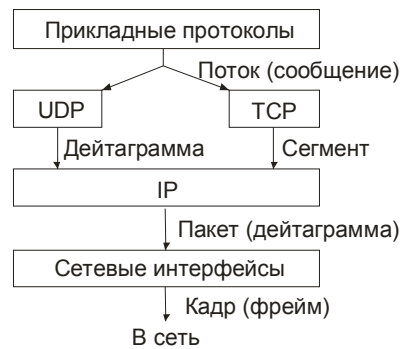


Рис. 3.12. Терминология в TCP/IP

Потоком данных, или просто **потоком (сообщением)**, называют данные, поступающие от приложений на вход протоколов транспортного уровня - TCP и UDP.

Протокол TCP «нарезает» из потока данных **сегменты**.

Единицу данных протокола UDP часто называют **дейтаграммой**, или **датаграммой**. Дейтаграмма — это общее название для единиц данных, которыми оперируют протоколы без установления соединений. К таким протоколам относится и протокол IP, поэтому его единицу данных также называют дейтаграммой. Однако очень часто используется и другой термин — **пакет**.

В стеке TCP/IP принято называть **кадрами**, или **фреймами**, единицы данных любых технологий, в которые упаковываются IP-пакеты для последующей переноски их через сети составной сети. При этом не имеет значения, какое название используется для этой единицы данных в технологии составляющей сети. Для TCP/IP фреймом является и кадр Ethernet, и ячейка ATM, и пакет X.25, так как все они выступают в качестве контейнера, в котором IP-пакет переносится через составную сеть.

Соответствие популярных стеков протоколов модели OSI

На рис 3.13 показано, в какой степени популярные стеки протоколов соответствуют рекомендациям модели OSI. Как мы видим, часто это соответствие весьма условно. В большинстве случаев разработчики стеков отдавали предпочтение скорости работы сети в ущерб модульности — ни один стек, кроме стека OSI, не разбит на семь уровней. Чаще всего в стеке явно выделяются 3-4 уровня: уровень сетевых адаптеров, в котором реализуются протоколы физического и канального уровней, сетевой уровень, транспортный уровень и уровень служб, вбирающий в себя функции сеансового уровня, уровня представления и прикладного уровня.

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

Модель OSI	IBM/ Microsoft	TCP/IP	Novell	Стек OSI
Прикладной	SMB	Telnet, FTP, SNMP, SMTP, WWW	NCP,	X.400 X.500 FTAM
Представления			SAP	Представ. протокол OSI
Сеансовый	NetBIOS			Сеансов. протокол OSI
Транспортный		TCP	SPX	Трансп. протокол OSI
Сетевой		IP, RIP, OSPF	IPX, RIP, NLSP	ES-ES IS-IS
Канальный	802.3 (Ethernet), 802.5 (Token Ring), FDDI, Fast Ethernet, SLIP, 100VG-AnyLAN, X.25, ATM, LAP-B, LAP-D, PPP			
Физический	Коаксиал, экранированная и неэкранированная витая пара, оптоволокно, радиоволны			

Рис.3.13. Соответствие популярных стеков модели OSI.

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

Информационные и транспортные услуги

Услуги компьютерной сети можно разделить на две категории:

- **транспортные услуги;**
- **информационные услуги.**

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

Информационные услуги состоят в предоставлении пользователю некоторой новой информации. Информационная услуга всегда связана с операциями по обработке информации: хранению ее в некотором упорядоченном виде (файловая система, база данных), поиску нужной информации и преобразованию информации.

Информационные услуги существовали и до появления первых компьютерных сетей, например, справочные услуги телефонной сети. С появлением компьютеров информационные услуги пережили революцию, так как компьютер и был изобретен для автоматической программной обработки информации. **Для оказания информационных услуг применяются различные информационные технологии: программирование, базы данных, файловые архивы, веб-сервис, электронная почта.**

В телекоммуникационных сетях «докомпьютерной» эры всегда преобладали **транспортные услуги**. Основной услугой телефонной сети всегда была передача голосового трафика между абонентами, в то время как справочные услуги были дополнительными.

В компьютерных сетях **одинаково важны обе категории услуг**. Эта особенность компьютерных сетей сегодня отражается на названии нового поколения телекоммуникационных сетей, которые появляются в результате конвергенции сетей различных типов. Такие сети все чаще стали называть **инфокоммуникационными**. Это название пока не стало общеупотребительным, но оно хорошо отражает новые тенденции, включая обе составляющие услуг на равных правах.

Деление услуг компьютерных сетей на две категории проявляется во многих аспектах. Существует, например, четкое деление специалистов в области компьютерных сетей на специалистов **информационных технологий и сетевых специалистов**:

1. **К первой категории** относятся программисты, разработчики баз данных, администраторы операционных систем, веб-дизайнеры, словом все те, кто имеет дело с разработкой и обслуживанием программного и аппаратного обеспечения компьютеров.
2. **Вторая категория** специалистов занимается транспортными проблемами сети. Эти специалисты имеют дело с каналами связи и коммуникационным оборудованием, таким как коммутаторы, маршрутизаторы и концентраторы. Они решают проблемы выбора топологии сети, выбора маршрутов для потоков трафика, определения требуемой пропускной способности каналов связи и коммуникационных устройств и другие проблемы, связанные только с передачей трафика через сеть.

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

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

Распределение протоколов по элементам сети

На рис. 3.14 показаны **основные элементы компьютерной сети**:

- **конечные узлы** — компьютеры
- **промежуточные узлы** — коммутаторы и маршрутизаторы (для примера выбраны протоколы стека TCP/IP, как наиболее распространенного).

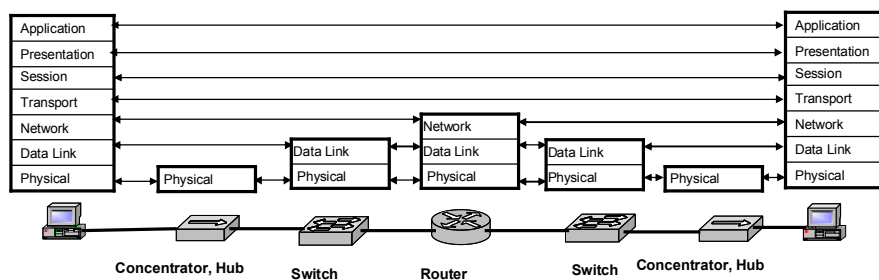


Рис. 3.14. Соответствие функций различных устройств сети уровням модели OSI

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.

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

Компьютеры, на которых работают сетевые приложения, должны поддерживать протоколы всех уровней. **Протоколы прикладного уровня, пользуясь сервисами протоколов уровня представления и сеансового уровня, предоставляют приложениям набор сетевых услуг в виде сетевого 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 функции канального уровня) реализуются программно операционной системой или системными приложениями.

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

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

Вспомогательные протоколы транспортной системы

На **рис. 3.15** показан упрощенный вариант распределения протоколов между элементами сети. В реальных сетях некоторые из коммуникационных устройств поддерживают не только протоколы трех нижних уровней, но и протоколы верхних уровней. Так, маршрутизаторы реализуют протоколы маршрутизации, позволяющие автоматически строить таблицы маршрутизации, а концентраторы и коммутаторы часто поддерживают протоколы SNMP и telnet, которые не нужны для выполнения основных функций этих устройств, но позволяют **конфигурировать и управлять ими удаленно.** Все эти протоколы являются протоколами прикладного уровня и выполняют некоторые вспомогательные (служебные) функции транспортной системы. Очевидно, что для работы прикладных протоколов сетевые устройства должны также поддерживать протоколы промежуточных уровней, таких как IP и TCP/UDP.

Вспомогательные протоколы можно разделить на группы, в соответствии с их функциями:

1. **Первую группу вспомогательных протоколов представляют протоколы маршрутизации**, такие как RIP, OSPF, BGP. Без этих протоколов маршрутизаторы не смогут продвигать пакеты, так как таблица маршрутизации будет пустой (если только администратор не заполнит ее вручную, но это не очень хорошее решение для крупной сети). Если рассматривать не только стек TCP/IP, но и стеки протоколов сетей с виртуальными каналами, то в эту группу попадают служебные протоколы, которые используются для установления виртуальных каналов.
2. **Другая группа вспомогательных протоколов выполняет преобразование адресов.** Здесь работает протокол DNS, который преобразует символьные имена узлов в IP-адреса. Протокол DHCP позволяет назначать IP-адреса узлам динамически, а не статически, что облегчает работу администратора сети.
3. **Третью группу образуют протоколы, которые используются для управления сетью.** В стеке TCP/IP здесь находится протокол SNMP (Simple Network Management Protocol — простой протокол управления сетью), который позволяет автоматически собирать информацию об ошибках и отказах устройств, а также протокол telnet, с помощью которого администратор может удаленно конфигурировать коммутатор или маршрутизатор.

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

И хотя такое деление отсутствует в модели OSI, оно существует в других стеках протоколов. Например, при стандартизации сетей ISDN, которые используют как принцип коммутации пакетов, так и принцип коммутации каналов, все протоколы разделены на три группы (рис. 3.15):

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

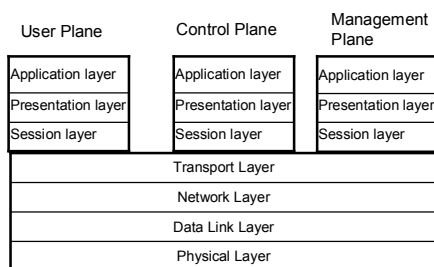


Рис. 3.15. Три группы протоколов

И хотя такое «горизонтальное» деление протоколов **пока не является общепринятым для компьютерных сетей, оно полезно**, так как позволяет глубже понять назначение протоколов. Кроме того, оно объясняет сложности при соотнесении некоторых протоколов уровням модели OSI. Например, в книгах одних авторов протоколы маршрутизации могут находиться на сетевом уровне, а в книгах других — на прикладном. Это происходит не из-за небрежности авторов, а из-за объективных трудностей классификации. **Модель OSI хорошо подходит для стандартизации протоколов, которые переносят пользовательский трафик**, то есть протоколов пользовательского слоя. В то же время она **в гораздо меньшей степени подходит для стандартизации вспомогательных протоколов**. Поэтому многие авторы и помещают протоколы маршрутизации на сетевой уровень, чтобы каким-то образом отразить функциональную близость этих протоколов к транспортным услугам сети, которые реализуются протоколом IP.

Выводы

Эффективной моделью средств взаимодействия компьютеров в сети является **многоуровневая структура**, в которой модули вышележащего уровня при решении своих задач рассматривают средства нижележащего уровня как некий инструмент. Каждый уровень данной структуры поддерживает интерфейсы двух типов:

- Во-первых, это **интерфейсы услуг с выше- и нижележащими уровнями** «своей» иерархии средств.
- Во-вторых, это **одноранговый интерфейс** со средствами другой взаимодействующей стороны, расположенными на том же уровне иерархии. Этот интерфейс называют **протоколом**.

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

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

Важнейшим направлением стандартизации в области вычислительных сетей является стандартизация коммуникационных протоколов. Примерами стандартизованных стеков протоколов являются TCP/IP, IPX/SPX, NetBIOS SMB, OSI, DECnet, SNA. Лидирующее положение занимает стек TCP/IP, он используется для связи десятков миллионов компьютеров всемирной информационной сети Интернет. Стек TCP/IP имеет 4 уровня: прикладной, транспортный, межсетевое взаимодействие и сетевых интерфейсов. Соответствие уровней стека TCP/IP уровням модели OSI достаточно условно.

Вопросы и задания

1. Что стандартизирует модель OSI?
2. Можно ли представить еще один вариант модели взаимодействия открытых систем с другим количеством уровней, например 8 или 5?
3. Ниже перечислены оригинальные (англоязычные) названия семи уровней модели OSI. Отметьте, какие из названий уровней не соответствуют стандарту?
 - physical layer
 - data link layer
 - network layer
 - transport layer
 - seances layer
 - presentation layer
 - application layer
4. Какие из приведенных утверждений вы считаете ошибочными:
 - протокол — это программный модуль, решающий задачу взаимодействия систем;
 - протокол — это формализованное описание правил взаимодействия, включающих последовательность обмена сообщениями и их форматы;
 - термины «интерфейс» и «протокол», в сущности, являются синонимами.
5. На каком уровне модели OSI работает прикладная программа?
6. Как вы считаете, протоколы транспортного уровня устанавливаются только на конечных узлах, только на промежуточном коммуникационном оборудовании (маршрутизаторах) или и там, и там?
7. На каком уровне модели OSI работают сетевые службы?
8. Ниже перечислены некоторые сетевые устройства:
 - маршрутизатор;
 - коммутатор;

- мост;
- повторитель;
- сетевой адаптер;
- концентратор.

В каком из этих устройств реализуются функции физического уровня модели OSI? Канального уровня? Сетевого уровня?

9. Какое название традиционно используется для единицы передаваемых данных на каждом из уровней? Заполните таблицу.

	Пакет	Сообщение	Кадр	Поток	Сегмент
Канальный уровень					
Сетевой уровень					
Транспортный уровень					
Сеансовый уровень					
Уровень представления					
Прикладной уровень					

- Какая организация стояла у истоков создания и стандартизации стека TCP/IP?
- Определите основные особенности стека TCP/IP.
- Сравните функции самых нижних уровней моделей TCP/IP и OSI.
- Дайте определение транспортных и информационных услуг.
- Какие протоколы относятся к слою управления (control plane)? А к слою менеджмента (management plane)?
- Должны ли маршрутизаторами поддерживаться протоколы транспортного уровня?
- Пусть на двух компьютерах установлено идентичное программное и аппаратное обеспечение за исключением того, что драйверы сетевых адаптеров Ethernet поддерживают отличающиеся интерфейсы с протоколом сетевого уровня IP. Будут ли эти компьютеры нормально взаимодействовать, если их соединить в сеть?
- Как организовать взаимодействие двух компьютеров, если у них отличаются протоколы:
 - физического и канального уровней;
 - сетевого уровня;
 - прикладного уровня.

ГЛАВА 4. Коммутация пакетов и каналов

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

Принцип **коммутации пакетов** был изобретен разработчиками компьютерных сетей. Коммутация пакетов учитывает особенности компьютерного трафика и является более эффективным способом коммутации для компьютерных сетей по сравнению с традиционным методом коммутации каналов, применяющимся в телефонных сетях.

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

Коммутация каналов

Чтобы объяснить основную идею коммутации каналов, рассмотрим ее в максимально упрощенном виде. Как показано на **рис. 4.1**, коммутационная сеть состоит из коммутаторов (S1, S2.....S5), связанных между собой **линиями связи**. Каждая линия имеет одну и ту же пропускную способность.

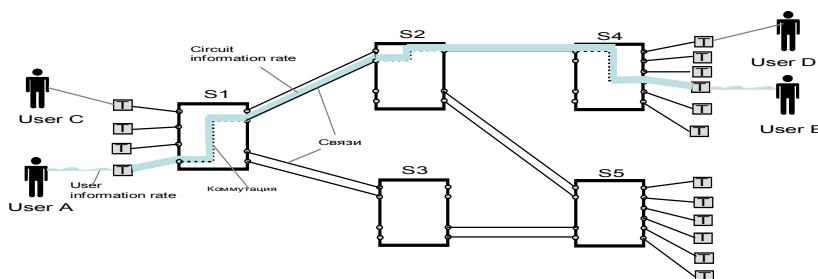


рис. 4.1. Коммутация каналов без мультиплексирования

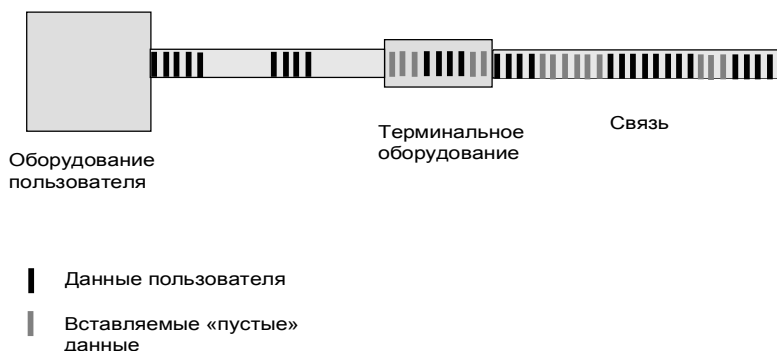


Рис. 4.2. Дополнение потока до пропускной способности линии

Каждый абонент подключается к сети с помощью терминального устройства (Т), которое посылает данные в сеть с постоянной скоростью, причем эта скорость в точности равна

пропускной способности линий. Если в какие-то периоды времени у абонента скорость информации, которую он хочет передать в сеть (предложенная нагрузка), оказывается меньше пропускной способности линии, то терминальное устройство продолжает питать сеть постоянным потоком данных, дополняя полезную информацию пользователя «пустыми» (незначущими) данными (рис. 4.2). О том, что часть потока битов несет полезную информацию, а часть является «заполнителем», знает терминальное устройство получателя, которое должно отбрасывать незначущую информацию и передавать пользователю только те данные, которые посылал в сеть отправитель.

Так как все мы - многолетние пользователи телефонной сети, которая является наиболее распространенным представителем сетей с коммутацией каналов, то мы будем сопровождать наше объяснение примерами из области *телефонии*.

Установление соединения

Обмен данными начинается с предварительного **установления соединения**.

Пусть два абонента **А** и **В** хотят передать друг другу некоторые данные (см. рис. 4.1). Прежде чем отправить данные в сеть (начать разговор), абонент **А** посылает **запрос** в коммутационную сеть, в котором указывается адрес (телефонный номер) абонента **В**. Цель посылки запроса — **установить соединение абонентов А и В информационным каналом**, свойства которого подобны свойствам непрерывной линии связи: на всем своем протяжении он передает данные с одной и той же скоростью. Это означает, что в транзитных коммутаторах **нет необходимости буферизовать данные** пользователей.

Для создания такого канала **запрос должен пройти через последовательность коммутаторов**, лежащих на пути от **А** к **В**. и **удостовериться, что все необходимые отрезки пути (линии связи) в данный момент свободны**. Кроме того, для успешного соединения необходимо, чтобы **конечный узел В не был занят в другом соединении**. Чтобы зафиксировать соединение, **в каждом из коммутаторов вдоль пути от А до В запоминается информация о том, что соответствующая линия связи выделена соединению абонентов А и В (зарезервирована)**. В каждом коммутаторе выполняется внутреннее соединение интерфейсов, соответствующих маршруту прохождения данных.

Отказ в установлении соединения

Важной особенностью технологии коммутации каналов является возможность **отказа в установлении соединения**. Если какая-либо другая пара абонентов обратится в сеть с запросом на установление соединения, для создания которого необходима хотя бы одна из уже назначенных для **А** и **В** линий связи, сеть ответит отказом. Например (см. рис. 4.1), если абонент **С** пошлет вызов в сеть на установление соединения с абонентом **Д**, то получит отказ, потому что единственная линия, соединяющая коммутаторы **S2** и **S4**, уже выделена соединению абонентов **А** и **В**. При отказе в установлении соединения сеть информирует вызывающего абонента об этом неприятном событии. Телефонная сеть в этом случае передает короткие гудки - сигнал «занято». Некоторые телефонные сети различают события «**сеть занята**» и «**абонент занят**», передавая гудки с разной частотой или используя разные тона.

Гарантированная пропускная способность

Итак, соединение абонентов **А** и **В** установлено. Теперь в их (и только в их) распоряжении имеется канал, обладающий фиксированной пропускной способностью. Это означает, что в течение всего времени соединения они должны посылать в сеть данные с этой скоростью и сеть гарантированно будет доставлять эти данные вызываемому абоненту без потерь и с той же скоростью вне зависимости от того, существуют ли и это время в сети другие соединения или нет. Абонент не может передавать данные в сеть со скоростью, превышающей

пропускную способность линии, но и сеть не может снижать скорость передачи пользовательских данных.

Нагрузка сети будет влиять только на вероятность отказа в установлении соединения – чем больше соединений в данный момент поддерживает сеть, тем больше вероятность того, что запрос на установление нового соединения столкнется с отказом

Положительным обстоятельством является то, что сеть будет **доставлять данные с низким и постоянным уровнем задержки**, что позволяет качественно передавать данные, чувствительные к задержкам (называемые также **трафиком реального времени**) - голос, видео.

Мультиплексирование

Описанная сеть с коммутацией каналов, в которой каждая **физическая линия всегда передает данные с одной и той же скоростью**, работает **неэффективно**.

Во-первых, пользователей в такой сети заставляют стать неким **универсальным стандартным пользователем**, который всегда передает в сеть информацию с постоянной единственно разрешенной скоростью. Сегодня трудно представить такого пользователя, учитывая оснащенность его различными терминальными устройствами — мобильными телефонами, компьютерами и офисными АТС (**корпоративный пользователь**). Поэтому скорость пользовательского трафика в общем случае **не совпадает с фиксированной пропускной способностью физических каналов**. Последняя может быть как намного выше, так и намного ниже потребности пользователя. В первом случае **пользователь недоиспользует канал**, а во втором должен либо **ограничить свои потребности**, либо **задействовать несколько физических линий связи**.

Во-вторых, и сама сеть в таком случае неэффективно использует собственные ресурсы. Очевидно, что в сети, изображенной на рис. 4.1. линий связи между коммутаторами недостаточно. Для того чтобы вероятность отказа была **достаточно низкой** и приемлемой для пользователя, между коммутаторами нужно проложить большое количество **параллельных физических линий**, а это очень затратный вариант.

Чтобы повысить эффективность в сетях с коммутацией каналов, стали использовать мультиплексирование; это позволило **одновременно передавать через каждый физический канал трафик нескольких логических соединений**.

Мультиплексирование в сетях с коммутацией каналов имеет свои особенности. Так, **пропускная способность каждой линии связи делится на равные части, образуя одинаковое число так называемых подканалов (для простоты их часто называют просто каналами)**. Обычно линия, подключающая пользователя к сети, поддерживает меньшее число подканалов, чем линии, соединяющие коммутаторы, — в этом случае вероятность отказа уменьшается. Например, пользовательская линия может состоять из 2, 24 или 30 подканалов, а линия между коммутаторами — из 480, 1920.

Наиболее распространенной скоростью цифрового подканала является сегодня **скорость 64 Кбит/с, которая обеспечивает качественную цифровую передачу голоса**.

После того как **сеть с коммутацией каналов была дополнена механизмом мультиплексирования**, схема ее работы претерпела некоторые изменения. Запрос на установление логического соединения, переданный абонентом перед началом передачи, **резервирует не целиком линии связи, а только их подканалы**. Таким образом, соединение устанавливается не на уровне линий связи, а на уровне подканалов. **Несколько подканалов используется в том случае, если пропускной способности одного канала оказывается недостаточно**. Это дает возможность пользователю **резервировать подканал (или подканалы)**, скорость передачи данных, которого является максимально близкой к реально необходимой. Кроме того,

мультиплексирование позволяет **более экономно строить связи между коммутаторами**. Действительно, для снижения вероятности отказов вместо нескольких линий связи теперь достаточно использовать одну физическую линию с большим количеством логических подканалов.

В случае **мультиплексирования в сети с коммутацией каналов** установленный при соединении составной канал по-прежнему состоит из линий связи с одинаковой пропускной способностью, только роль линий связи играют подканалы.

Очевидно, что при мультиплексировании усложняется процедура обработки трафика коммутаторами — вместо простой и понятной процедуры коммутации соответствующих интерфейсов теперь необходимо передать данные в нужный подканал. При **временном мультиплексировании** это требует высокой степени синхронизации двух информационных потоков, а при **частотном** — преобразования частот.

На **рис.4.3** показана сеть с коммутацией каналов и мультиплексированием. В сети установлены два соединения **A-B** и **C-D**, одно из них использует **по одному подканалу в каждой линии связи**, а второе — **по два**. Таким образом, несмотря на то, что данная сеть имеет ту же физическую структуру, что и сеть, представленная на рис. 4.1, **блокирования второго вызова (C-D) здесь не происходит**, так как коммутаторы поддерживают мультиплексирование.

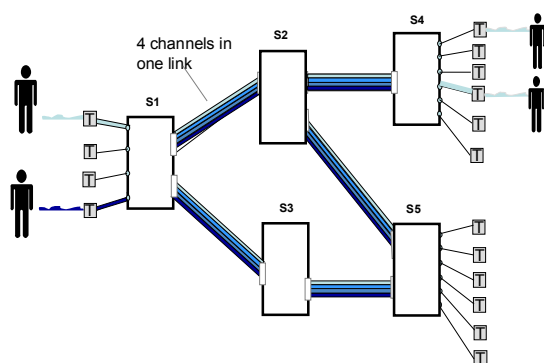


рис. 4.3. Коммутация каналов с мультиплексированием

Неэффективность передачи пульсирующего трафика

Имеется еще одна причина неэффективной работы сети с коммутацией каналов, а именно: **фиксированная пропускная способность составного канала выделяется на все время соединения**.

Мы отметили, что мультиплексирование повышает эффективность работы сети, потому что пользователь может точнее подобрать скорость соединения в соответствии со своими реальными потребностями. Но это относится **только к пользователям, скорость информационных потоков которых постоянна**.

Однако, если присмотреться к пользовательскому трафику, то окажется, что **практически все пользователи телекоммуникационных сетей не относятся к этой категории**. Так, кажущееся постоянство скорости информационных потоков пользователей телефонной сети достигается за счет обработки **неравномерных пользовательских информационных потоков терминальными устройствами телефонной сети — телефонами**. Например, цифровой телефон действительно постоянно передает в сеть информационный поток со скоростью 64 Кбит/с независимо от того, говорит в это время его пользователь или нет. Наверно, более эффективно телефон работал бы, если бы он **«вырезал»** паузы из разговора и передавал в сеть только полезную информацию.

Существует и другая категория пользователей, потребности которой в передаче информации с различной скоростью еще более очевидны, — это **пользователи компьютеров**. Действительно, работа пользователя, который сканирует веб-ресурсы Интернета, очевидным образом порождает **неравномерный трафик**. При загрузке очередной страницы в компьютер скорость резко возрастает, а после окончания загрузки падает практически до нуля.

Коэффициент пульсации трафика отдельного пользователя сети определяется как **отношение пиковой скорости на каком-либо небольшом интервале времени к средней скорости обмена данными на длительном интервале времени и может достигать значений 100:1**.

Если для описанного сеанса организовать коммутацию канала между компьютером пользователя и сервером, то большую часть времени канал будет простаивать. В то же время часть производительности сети останется закрепленной за данной группой абонентов и недоступной другим пользователям сети. Сеть в такие периоды похожа на пустой эскалатор метро, который движется, но полезную работу не выполняет, другими словами, «перевозит воздух».

Наиболее эффективно пользовательский трафик передается сетью с коммутацией каналов в том случае, когда скорость его постоянна в течение всего сеанса связи и соответствует пропускной способности физических линий связи.

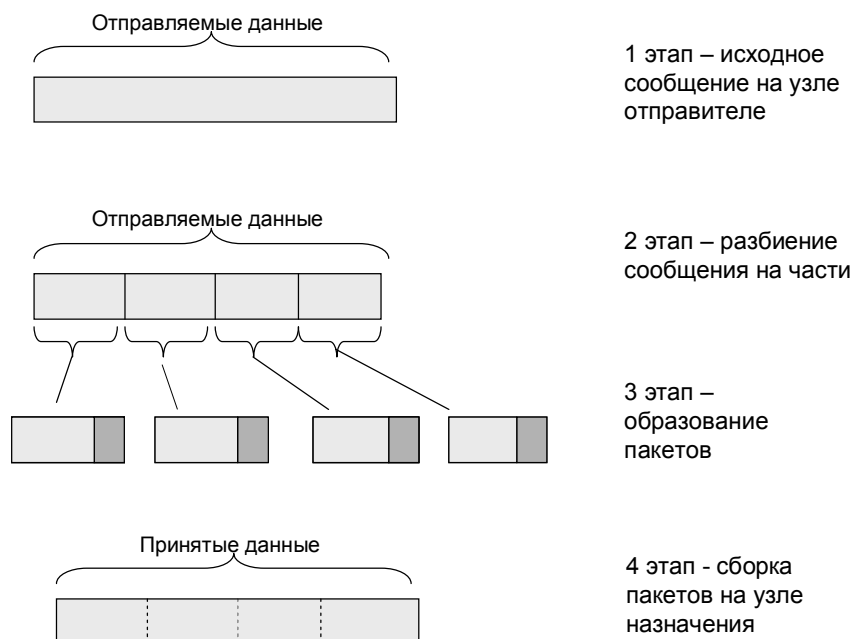
Достоинства и недостатки любой сетевой технологии — относительны. В определенных ситуациях на первый план выходят достоинства, а недостатки становятся незначительными. Так, **техника коммутации каналов** хорошо работает в тех случаях, когда нужно передавать трафик **телефонных разговоров**, а с невозможностью «вырезать» паузы из разговора и нерациональным использованием магистральных физических каналов между коммутаторами можно мириться. Однако при передаче очень **неравномерного компьютерного трафика** на первый план выходит именно эта **нерациональность**.

Коммутация пакетов

Техника коммутации пакетов была специально разработана для эффективной передачи компьютерного трафика.

При коммутации пакетов все передаваемые пользователем сети данные разбиваются в исходном узле на сравнительно небольшие части, называемые **пакетами, кадрами, или ячейками**, — в данном контексте различия в значении этих терминов не существенны (рис. 4.4). Каждый пакет снабжается **заголовком**, в котором указывается **адрес**, необходимый для доставки пакета узлу назначения. Наличие адреса в каждом пакете является одним из важнейших свойств техники коммутации пакетов, так как **каждый пакет может быть обработан коммутатором независимо от других пакетов информационного потока**. Помимо заголовка у пакета имеется еще одно дополнительное поле, которое обычно размещается в конце пакета и поэтому называется **концевиком**. В концевике помещается **контрольная сумма**, которая позволяет проверить, была ли искажена информация при передаче через сеть или нет.

Рис. 4.4. Разбиение потока данных на пакеты



Буферы и очереди

Сеть с коммутацией пакетов, так же как и сеть с коммутацией каналов, состоит из коммутаторов, связанных физическими линиями связи. Однако коммутаторы функционируют в этих сетях по-разному. Главное отличие состоит в том, что пакетные коммутаторы имеют внутреннюю буферную память для временного хранения пакетов.

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

Буферизация необходима пакетному коммутатору также для согласования скорости поступления пакетов со скоростью их коммутации. Если коммутирующий блок не успевает обрабатывать пакеты, то на интерфейсах коммутатора возникают входные очереди. Очевидно, что для хранения входной очереди объем буфера должен превышать размер одного пакета.

Существуют различные подходы к построению коммутирующего блока:

1. Традиционный способ основан на **одном центральном процессоре**, который обслуживает все входные очереди коммутатора. Такой способ построения может приводить к большим очередям, так как производительность процессора разделяется между несколькими очередями.
2. Современные способы построения коммутирующего блока основаны на **многопроцессорном подходе**, когда каждый интерфейс имеет свой встроенный процессор для обработки пакетов. Кроме того, существует также центральный процессор, координирующий работу интерфейсных процессоров. Использование интерфейсных процессоров повышает производительность коммутатора и уменьшает очереди во входных интерфейсах. Однако такие очереди все равно могут возникать, так как центральный процессор по-прежнему остается «узким местом»..

Наконец, буферы нужны для согласования скоростей передачи данных в каналах, подключенных к пакетному коммутатору. Действительно, если скорость поступления

пакетов из одного канала в течение некоторого периода превышает пропускную способность того канала, в который эти пакеты должны быть направлены, то во избежание потерь пакетов на целевом интерфейсе необходимо организовать выходную очередь (рис. 4.5).

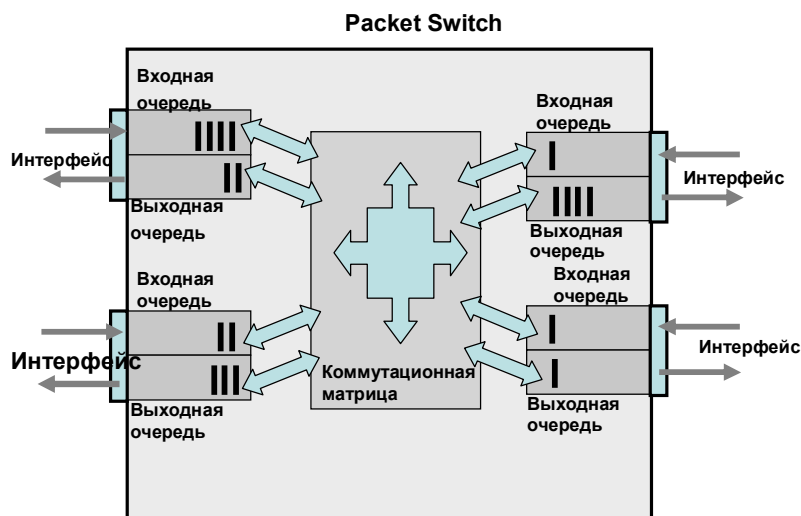


Рис.4.5. Очереди в пакетном коммутаторе

В сети с коммутацией пакетов пульсации трафика отдельных абонентов в соответствии с законом больших чисел распределяются во времени так, что их пики чаще всего не совпадают. Поэтому коммутаторы постоянно и достаточно равномерно загружены работой, если число обслуживаемых ими абонентов действительно велико.

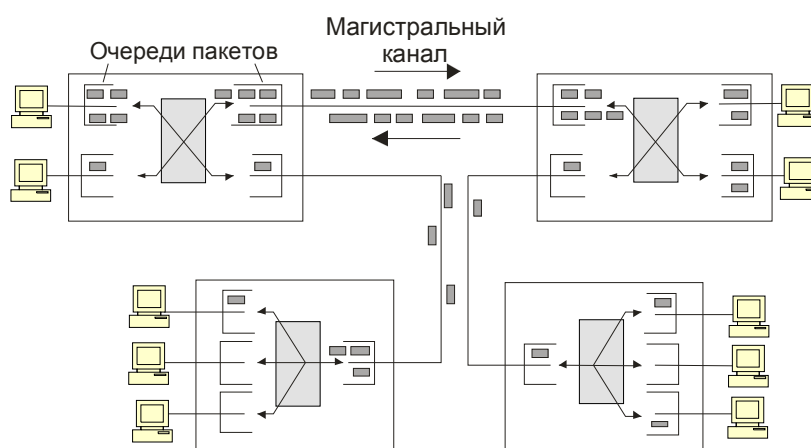


Рис. 4.6. Сглаживание пульсаций трафика в сети с коммутацией пакетов

На рис. 4.6 показано, что трафик, поступающий от каждого из конечных узлов на коммутаторы, очень **неравномерно распределен во времени**. Однако коммутаторы более высокого уровня иерархии, которые обслуживают соединения между коммутаторами нижнего уровня, загружены более равномерно, и **магистральные каналы, соединяющие коммутаторы верхнего уровня, имеют близкие к максимальным коэффициенты использования**. Буферизация сглаживает пульсации, поэтому коэффициент пульсации на магистральных каналах гораздо ниже, чем на каналах абонентского доступа.

Поскольку объем буферов в коммутаторах ограничен, иногда происходит потеря пакетов из-за **переполнения** буферов при временной **перегрузке** части сети, когда **совпадают периоды пульсации нескольких информационных потоков**. Так как потеря пакетов является неотъемлемым свойством сети с коммутацией пакетов, то для нормальной работы таких сетей разработан ряд механизмов, которые компенсируют этот эффект. **Эти механизмы, называются методами обеспечения качества обслуживания (QoS)**.

Методы продвижения пакетов

Решение о том, на какой интерфейс передать пришедший пакет, принимается на основании одного из трех методов продвижения пакетов:

- При **дейтаграммной передаче** соединение не устанавливается, и все передаваемые пакеты **продвигаются (передаются от одного узла сети другому) независимо** друг от друга на основании одних и тех же правил. Процедура обработки пакета определяется только значениями параметров, которые он несет в себе, и текущим состоянием сети (например, в зависимости от ее нагрузки пакет может стоять в очереди на обслуживание большее или меньшее время). Однако никакая информация об уже переданных пакетах сетью не хранится и в ходе обработки очередного пакета во внимание не принимается. То есть каждый отдельный пакет рассматривается сетью как совершенно независимая единица передачи — **дейтаграмма**.
- **Передача с установлением логического соединения** распадается на так называемые сеансы, или логические соединения. Процедура обработки определяется не для отдельного пакета, а для всего множества пакетов, передаваемых в рамках каждого соединения. Для того чтобы реализовать дифференцированное обслуживание пакетов, принадлежащих разным соединениям, сеть должна, **во-первых**, присвоить каждому соединению **идентификатор**, **во-вторых**, запомнить параметры соединения, то есть значения, определяющие процедуру обработки пакетов в рамках данного соединения. Эта информация называется **информацией о состоянии соединения**. Фиксированный маршрут не является обязательным параметром соединения. Пакеты, принадлежащие одному и тому же соединению, то есть имеющие одни и те же адреса отправления и назначения, могут перемениться по разным независимым друг от друга маршрутам.
- **Передача с установлением виртуального канала**. Если в число параметров соединения **входит маршрут**, то все пакеты, передаваемые в рамках данного соединения, должны проходить по указанному пути. Такой единственный заранее проложенный фиксированный маршрут, соединяющий конечные узлы в сети с коммутацией пакетов, называют **виртуальным каналом** (virtual circuit, или virtual channel).

Классификация методов коммутации приведена на **рис. 4.7**.

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

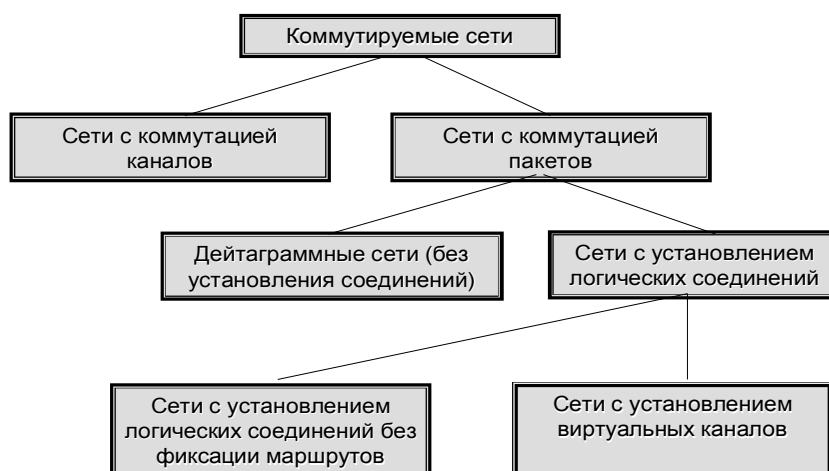


Рис. 4.7. Классификация коммутируемых сетей

Дейтаграммная передача

Дейтаграммный способ передачи данных основан на том, что **все передаваемые пакеты обрабатываются независимо друг от друга**. Выбор интерфейса, на который надо передать поступивший пакет, происходит только на основании **адреса назначения**, содержащегося в заголовке пакета. **Принадлежность пакета к определенному информационному потоку никак не учитывается.**

Решение о продвижении пакета принимается на основе **таблицы коммутации, содержащей набор адресов назначения и адресную информацию, определяющую следующий по маршруту (транзитный или конечный) узел.** В разных технологиях для обозначения таблиц, имеющих указанное выше функциональное назначение, могут использоваться другие термины (таблица маршрутизации, таблица продвижения и др.). Далее для простоты будем пользоваться термином «таблица коммутации» в качестве обобщенного названия таблиц такого рода, применяемых для **дейтаграммной передачи на основании только адреса назначения конечного узла.**

Таблица коммутации дейтаграммной сети должна содержать записи обо всех адресах, куда могут быть направлены пакеты, поступающие на интерфейсы коммутатора. А они в общем случае могут быть адресованы любому узлу сети. На практике используются приемы, уменьшающие число записей в таблице, например, **иерархическая адресация.** В этом случае таблица коммутации может содержать только старшие части адресов, которые соответствуют не отдельным узлам, а некоторой группе узлов (для их обозначения часто применяют термин «**подсеть**»).

Несмотря на применение иерархической адресации в некоторых крупных сетях (например, в Интернете), коммутаторы могут иметь таблицы с числом адресов назначения, превышающем несколько тысяч. На **рис. 4.8** показано, как могла бы выглядеть таблица коммутации в дейтаграммной сети.

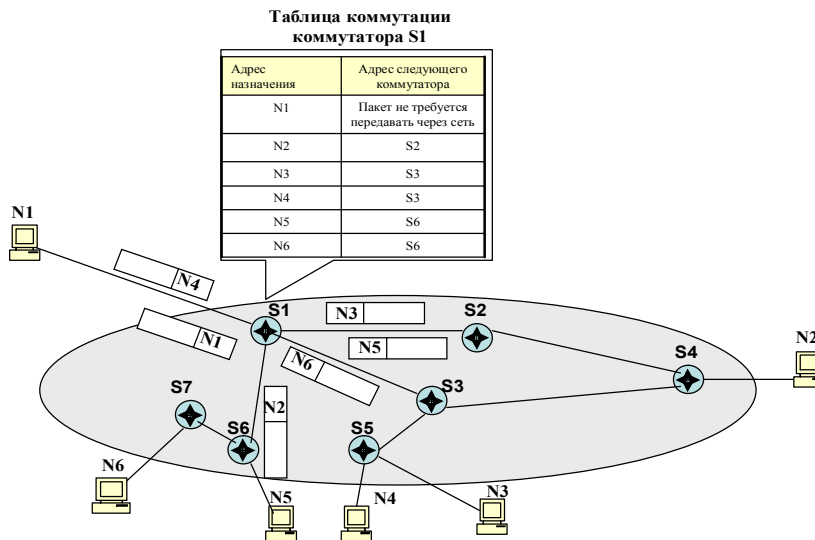


Рис. 4.8. Дейтаграммный принцип передачи пакетов

В таблице коммутации для одного и того же **адреса назначения может содержаться несколько записей**, указывающих соответственно на различные адреса следующего коммутатора. Такой подход называется **балансом нагрузки и используется для повышения производительности и надежности сети**. В примере, показанном на рис. 4.8, пакеты, поступающие в коммутатор S1 для узла назначения с адресом N2 в целях баланса нагрузки распределяются между двумя следующими коммутаторами - S2 и S3, что снижает нагрузку на каждый из них, а значит, уменьшает очереди и ускоряет доставку. Некоторая «размытость» путей следования пакетов с одним и тем же адресом назначения через сеть является прямым следствием принципа независимой обработки каждого пакета, присущего дейтаграммному методу. **Пакеты, следующие по одному и тому же адресу назначения, могут добираться до него разными путями также вследствие изменения состояния сети, например отказа промежуточных коммутаторов.**

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

Логическое соединение

Передача с **установлением логического соединения** основывается на знании «**предыстории**» обмена. Это позволяет **более рационально** по сравнению с дейтаграммным способом обрабатывать пакеты. Например, при потере нескольких предыдущих пакетов может быть снижена скорость отправки последующих. Или благодаря нумерации пакетов и отслеживанию номеров отправленных и принятых пакетов **можно повысить надежность** путем отбрасывания дубликатов, упорядочивания поступивших и повторения передачи потерянных пакетов.

Параметры соединения могут быть как **постоянными** в течение всего соединения (например, максимальный размер пакета), так и **переменными**, динамически отражающими текущее состояние соединения (например, последовательные номера пакетов). Когда **отправитель и получатель фиксируют начало нового соединения, они, прежде всего, «договариваются» о начальных значениях параметров процедуры обмена** и только после этого начинают передачу собственно данных.

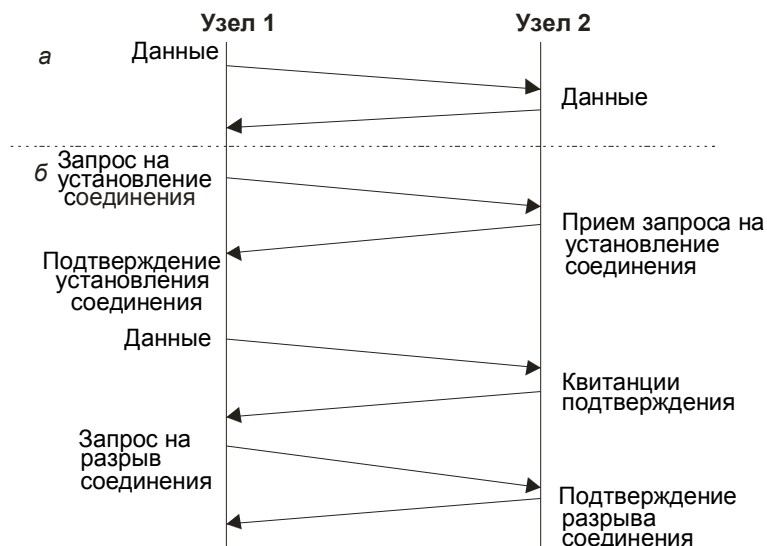


Рис. 4.9. Передача без установления соединения (а). Передача с установлением соединения (б)

Передача с установлением соединения более надежна, но требует больше времени для передачи данных и вычислительных затрат от конечных узлов, что иллюстрирует **рис. 4.9.б**.

При передаче с установлением соединения узлу-получателю отправляется **служебный кадр** специального формата с предложением установить соединение, как показано на **рис. 4.9. б**. Если узел-получатель согласен с этим, то он посылает в ответ **другой служебный кадр**, подтверждающий установление соединения и предлагающий некоторые параметры, которые будут использоваться в рамках данного логического соединения. Это могут быть, например, **идентификатор соединения, максимальное значение длины поля данных кадров, количество кадров, которые можно отправить без получения подтверждения, и т. п.** Узел-инициатор соединения может закончить процесс установления соединения отправкой третьего служебного кадра, в котором сообщит, что предложенные параметры ему подходят. На этом логическое соединение считается установленным.

Логическое соединение может быть рассчитано на передачу данных как в одном направлении — от инициатора соединения, так и в обоих направлениях. После передачи некоторого законченного набора данных, например определенного файла, узел-отправитель **инициирует разрыв данного логического соединения**, посылая соответствующий служебный кадр.

В отличие от передачи дейтаграммного типа, в которой поддерживается только один тип кадра — **информационный**, передача с установлением соединения должна поддерживать как минимум два типа кадров — **информационные**, переносящие собственно пользовательские данные, и **служебные**, предназначенные для установления (разрыва) соединения.

Виртуальный канал

Виртуальные каналы (virtual circuit, или virtual channel) — это **устойчивые пути следования трафика, создаваемые в сети с коммутацией пакетов**. Виртуальные каналы являются базовой концепцией технологий **X.25, Frame Relay и ATM**.

Техника виртуальных каналов **учитывает существование в сети потоков данных**. Для того чтобы выделить поток данных из общего трафика, каждый пакет этого потока помечается

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

Эта метка в различных технологиях называется по-разному: **номером логического канала** (Logical Channel number, LCN) в технологии X.25, **идентификатором соединения уровня канала данных** (Data Link Connection Identifier, DLCI) в технологии Frame Relay, **идентификатором виртуального канала** (Virtual Channel Identifier, VCI) в технологии ATM.

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

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

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

На рис. 4.10 показан фрагмент сети, в которой проложены два виртуальных канала - VC1 и VC2.

Первый проходит от конечного узла с адресом **N1**, до конечного узла с адресом **N2** через промежуточные коммутаторы **S1, S2 и S4**. Второй обеспечивает продвижение данных по пути **N3 — S5 — S3 — S1 — N1**. В принципе между двумя конечными узлами может быть проложено несколько виртуальных каналов, например, еще один виртуальный канал между узлами **N1 и N2** мог бы проходить через промежуточные коммутаторы **S1, S3 и S4**.

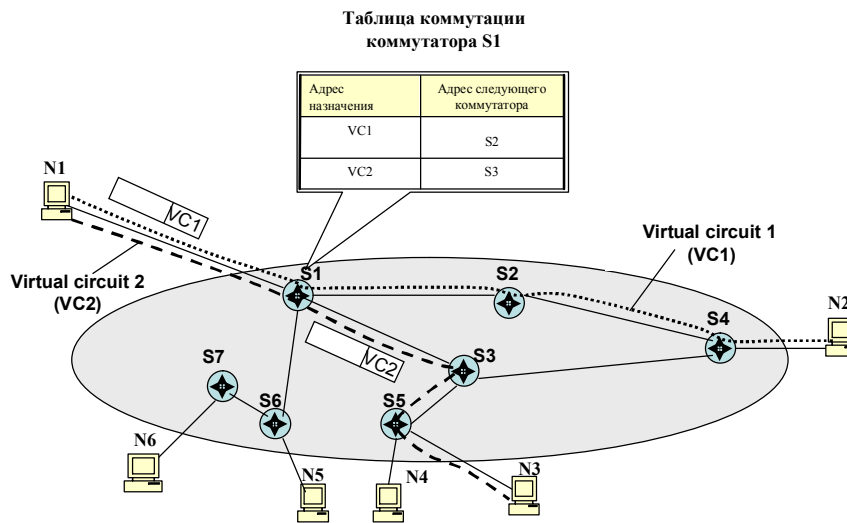


Рис. 4.10. Иллюстрация принципа работы виртуального канала

Таблица 3.1. Сравнение сетей с коммутацией каналов и пакетов

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

Разделение среды

Разделяемой средой называется физическая среда передачи данных, к которой непосредственно подключено несколько конечных узлов сети. Причем в каждый момент времени только один из конечных узлов получает доступ к разделяемой среде и задействует ее для передачи пакета (или нескольких пакетов) другому узлу, подключенному к этой же среде.

В качестве разделяемой среды может использоваться коаксиальный кабель, витая пара, оптическое волокно или радиоволны.

Одним из возможных подходов к разделению среды является **метод случайного доступа**. При методе случайного доступа может возникать ситуация, когда одновременно два или более компьютеров решают, что сеть свободна, и начинают передавать информацию. Такая ситуация, называемая **коллизией**, препятствует правильной передаче данных по сети. Сигналы нескольких передатчиков накладываются друг на друга, и суммарный сигнал становится искаженным.

Во всех сетевых технологиях, использующих метод случайного доступа, предусмотрен алгоритм обнаружения и корректной обработки коллизий. Вероятность возникновения коллизии зависит от **интенсивности** сетевого трафика.

После **обнаружения коллизии** сетевые адаптеры, которые пытались передать свои кадры, прекращают передачу и после паузы случайной длительности пытаются снова получить доступ к среде, чтобы передать кадр, вызвавший коллизию.

Управление доступом к среде в данном случае осуществляется децентрализованно: в этом процессе участвуют все сетевые интерфейсы, непосредственно подключенные к разделяемой среде. В частности, в компьютерах эту функцию выполняют **сетевые интерфейсные карты**, или **сетевые адаптеры**.

Суть случайного метода доступа состоит в следующем:

1. Компьютер может передавать данные по сети, только если сеть свободна, то есть если никакой другой компьютер в данный момент не занимается обменом и электрические (или оптические) сигналы в среде отсутствуют.
2. После того как компьютер убеждается, что среда свободна, он начинает передачу, «захватывая» среду. Время монопольного использования разделяемой среды одним узлом ограничивается временем передачи одного кадра.
3. При попадании кадра в разделяемую среду все сетевые адаптеры одновременно начинают принимать этот кадр. Каждый из них анализирует адрес назначения, располагающийся в одном из начальных полей кадра.
4. Если этот адрес совпадает с их собственным адресом, кадр помещается во внутренний буфер сетевого адаптера. Таким образом компьютер-адресат получает предназначенные ему данные.

Другим распространенным способом доступа к разделяемой среде является **метод детерминированного доступа**. Этот метод основан на использовании кадра специального формата, который обычно называют **маркером**, или **токеном**, доступа. Компьютер имеет право пользоваться разделяемой средой только тогда, когда он владеет токеном. Время владения токеном ограничено, так что после истечения этого срока компьютер обязан передать токен другому компьютеру. Правило, определяющее порядок передачи токена, должно гарантировать каждому компьютеру доступ к разделяемой среде в течение некоторого фиксированного времени.

Метод детерминированного доступа может быть реализован на основе **как распределенного, так и централизованного подходов**. В первом случае в сети нет узла, определяющего очередность владения разделяемой средой, во втором - такой узел существует и называется **арбитром доступа**.

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 (имеют одинаковые права) 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, маршрутизаторов.

Физическая структуризация локальной сети

Различают топологию физических связей (**физическую структуру сети**) и топологию логических связей сети (**логическую структуру сети**).

Конфигурация **физических связей** определяется электрическими соединениями компьютеров и может быть представлена в виде геометрической фигуры, узлами которого являются компьютеры и коммуникационное оборудование, а ребра соответствуют отрезкам кабеля, связывающим пары узлов

Логические связи представляют собой пути прохождения информационных потоков по сети; они образуются путем соответствующей настройки коммуникационного оборудования.

В некоторых случаях **физическая и логическая топологии сети совпадают**. Например, сеть, представленная на **рис. 4.11, а**, имеет физическую кольцевую топологию. Пусть компьютеры этой сети используют метод детерминированного доступа. Причем **маркер** всегда передается последовательно от компьютера к компьютеру в том же порядке, в котором компьютеры образуют физическое кольцо: то есть компьютер А передает **маркер** компьютеру В, компьютер В — компьютеру С и т. д. В этом случае логическая топология сети также является кольцом.

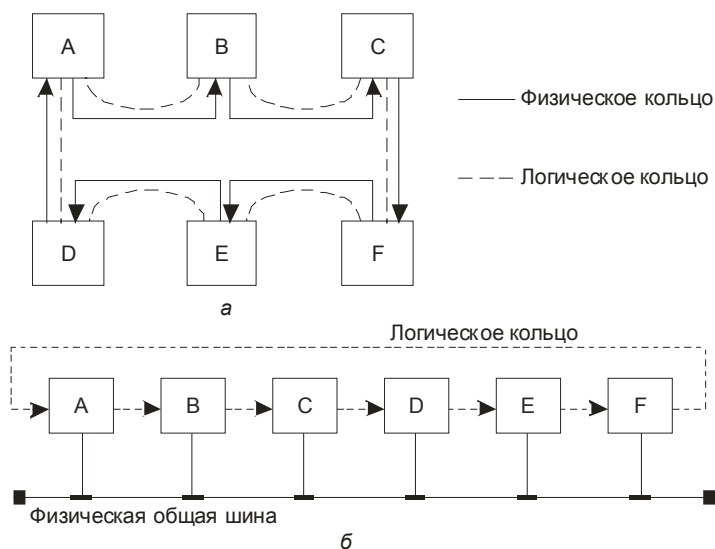


Рис. 4.11. Логическая и физическая топологии сети

Сеть, показанная на рис. 4.11. б, является примером несовпадения физической и логической топологий. Физически компьютеры соединены по топологии общей шины (звезда). Доступ же к шине происходит не по алгоритму случайного доступа, а путем передачи маркера в кольцевом порядке: от компьютера А — компьютеру В, от компьютера В — компьютеру С и т. д. Здесь порядок передачи маркера 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: В. А. С... 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.

Основными средствами физической структуризации локальных сетей являются **повторители (repeater)** и **концентраторы (concentrator)**, или **хабы (hub)**

Простейшее из коммуникационных устройств — **повторитель (repeater)** — используется для **физического соединения различных сегментов кабеля локальной сети с целью увеличения общей длины сети и количества компьютеров в сети**. Повторитель повторяет сигналы, приходящие из одного сегмента сети в другие ее сегменты (рис. 4.12), 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 и количество компьютеров**. Так как поток сигналов, передаваемых узлом в сеть, распространяется по всем отрезкам сети, такая сеть остается **сетью с единой разделяемой средой**.

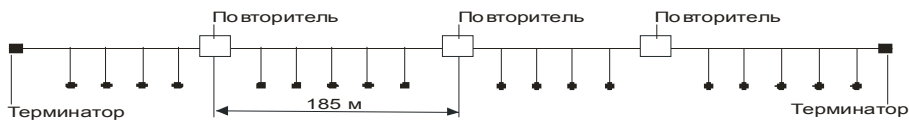


Рис. 4.12. Повторители позволяют увеличить длину сети Ethernet

Повторитель, который имеет несколько портов и соединяет несколько физических сегментов, часто называют **концентратором**, или **хабом**. Эти названия отражают тот факт, что в данном устройстве **сосредотачиваются все связи между сегментами сети**.

Добавление в сеть концентратора 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, 100VG-AnyLAN**. Не все из перечисленных технологий актуальны — технологии **ArcNet** и **100VG-AnyLAN** могут рассматриваться сегодня только в качестве примеров оригинальных технических решений

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. 4.13, а**). А концентратор Token Ring (**рис. 4.13, б**) повторяет входной сигнал только на *одном, соседнем* порту.

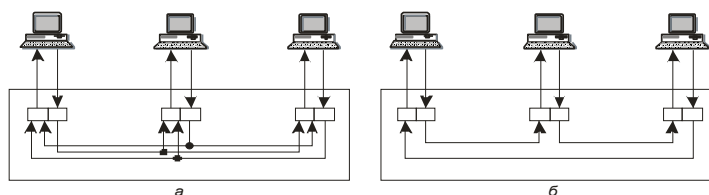


Рис. 4.13. Концентраторы различных технологий

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 непохожими структуре информационных потоков в большой сети.

Пример

Пусть на предприятии имелась простейшая односегментная сеть Ethernet (рис. 4.14. а). К коаксиальному кабелю были подключены все имеющиеся на предприятии компьютеры. Шло время, **количество пользователей увеличивалось**, сеть **все чаще оказывалась занятой**, пользователям приходилось все дольше ждать ответа сетевых приложений. Кроме того, **начали сказываться ограничения на длину связей между компьютерами**, так оказалось невозможным разместить компьютеры в помещении, выделенном для новой рабочей группы. Было принято решение применить концентраторы.

На рис. 4.14. б показана сеть, **полученная в результате физической структуризации**. Появилась возможность разнести компьютеры пользователей на большие расстояния, и **физическая структура сети стала соответствовать административному устройству предприятия**. Однако проблемы, связанные с производительностью, остались нерешенными. Например, каждый раз, когда пользователь компьютера **А** посылал данные своему соседу, работающему за компьютером **В**, оказывалась блокированной вся сеть. Ничего удивительного — в соответствии с логикой работы концентратора кадр, посылаемый компьютером **А** компьютеру **В**, повторялся на всех интерфейсах всех узлов сети. И до тех пор, пока компьютер **В** не получал адресованный ему кадр, ни один из компьютеров сети не мог получить доступ к разделяемой среде передачи. **Такая ситуация возникла из-за того, что использование концентраторов изменило только физическую структуру сети, оставив без изменения ее логическую структуру.**



Рис. 3.18. Изменение физической структуры сети не сказывается на производительности сети

Рис. 4.14

Решение представленной в примере проблемы состоит в **отказе от идеи одной общей для всех узлов разделяемой среды**. Например, в данном случае желательно было бы сделать так, чтобы кадры, которые перелают компьютеры отдела 1, выхолили бы за пределы этой части сети в том и только в том случае, если они направлялись компьютеру какого-либо из других

отделов. **С другой стороны**, в сеть каждого из отделов должны попадать те и только те кадры, которые адресованы узлам этой сети. **Таким образом, в пределах каждого отдела используется отдельная «собственная» разделяемая среда.**

Распространение трафика, предназначенного для компьютеров некоторого сегмента сети, только в пределах этого сегмента называется локализацией трафика. Логическая структуризация сети – это процесс разбиения сети на сегменты с локализованным трафиком.

При правильно проведенной логической структуризации производительность сети может существенно повыситься, так как **компьютеры одного отдела не будут простаивать в то время, когда обмениваются данными компьютеры других отделов.** Кроме того, логическая структуризация позволяет дифференцировать доступную пропускную способность в разных частях сети.

Логическая структуризация сети проводится путем использования мостов, коммутаторов, маршрутизаторов и шлюзов.

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 (рис. 4.15). Тем самым мост **изолирует трафик одного сегмента от трафика другого, повышая общую производительность сети.** Локализация трафика не только экономит пропускную способность, но и снижает **возможность несанкционированного доступа к данным**, так как кадры не выходя за пределы своего сегмента и их сложнее перехватить злоумышленнику.

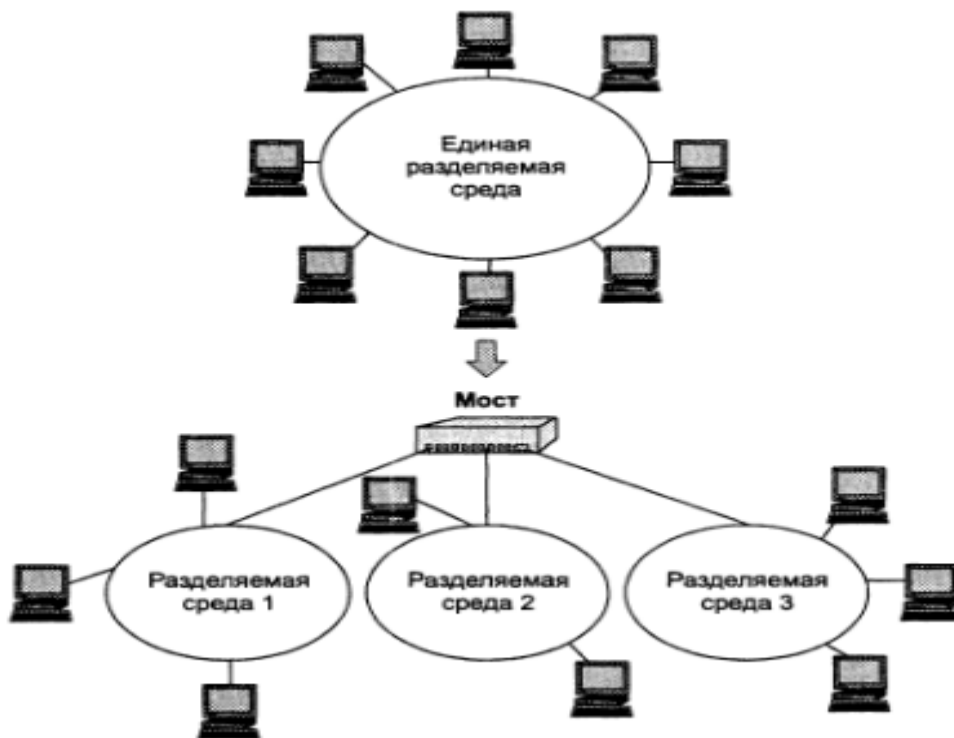


Рис. 3.19. Мост делит единую среду передачи

Рис. 4.15

На рис. 4.16 показана сеть, которая была получена из сети с центральным концентратором (см. рис. 4.14. б) путем его замены мостом. Сети отделов 1 и 2 состоят из отдельных логических сегментов, а сеть отдела 3 — из двух логических сегментов. Каждый логический сегмент построен на базе концентратора и имеет простейшую физическую структуру, образованную отрезками кабеля, связывающими компьютеры с портами

концентратора. Если пользователь компьютера А пошлет данные пользователю компьютера В, находящемуся в одном с ним сегменте, то эти данные будут повторены только на сетевых интерфейсах их общего сегмента.

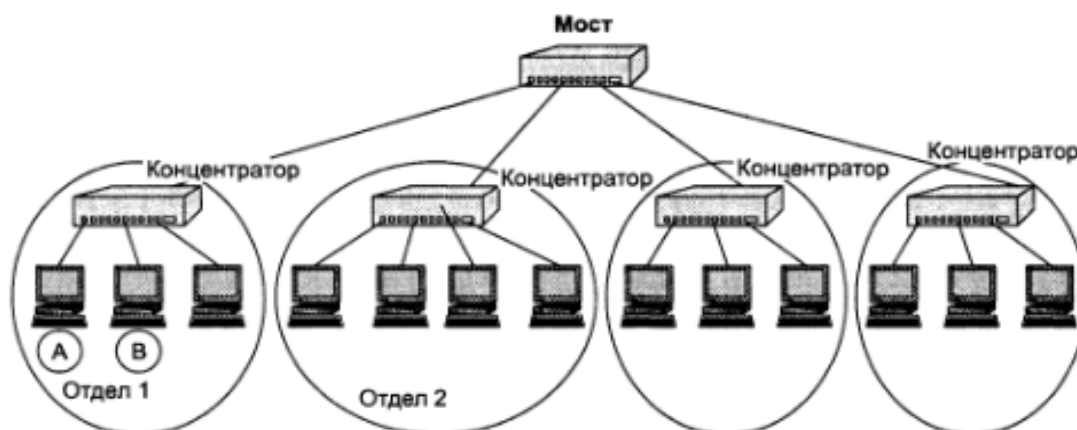


Рис. 3.20. Единая разделяемая среда с помощью моста преобразована в четыре разделяемых среды

Рис.4.16.

Мосты используют для локализации трафика **аппаратные адреса компьютеров**. Каким образом мост узнает интерфейс, на который надо передать кадр, ведь аппаратный адрес не содержит никакой информации о принадлежности компьютера с данным адресом тому или иному сегменту. Такая информация может быть предоставлена мосту **администратором в результате ручного конфигурирования**. Однако такой способ мало пригоден для больших сетей.

Мост решает эту задачу автоматически, используя простой обучающий алгоритм. Все кадры, приходящие на определенный интерфейс моста, сгенерированы компьютерами, относящимися к сегменту, подключенному к этому интерфейсу. Мост **извлекает из поступающих кадров адреса отправителей** и помещает их в таблицу, где делает отметку о том, **на какой его интерфейс поступил каждый из кадров**. Так мост определяет, какие компьютеры подключены к каждому из его интерфейсов. В дальнейшем мост использует эту информацию для передачи кадра именно на тот интерфейс, через который идет путь к компьютеру назначения. Поскольку, точная топология связей между логическими сегментами мосту неизвестна, он может правильно работать только в тех сетях, в которых межсегментные связи **не образуют замкнутых контуров (петель)**.

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.

Рассмотрим ограничение функциональных возможностей мостов и коммутаторов из-за отсутствия поддержки петлеобразных конфигураций на примере сети показанной на **рис. 4.16а**.

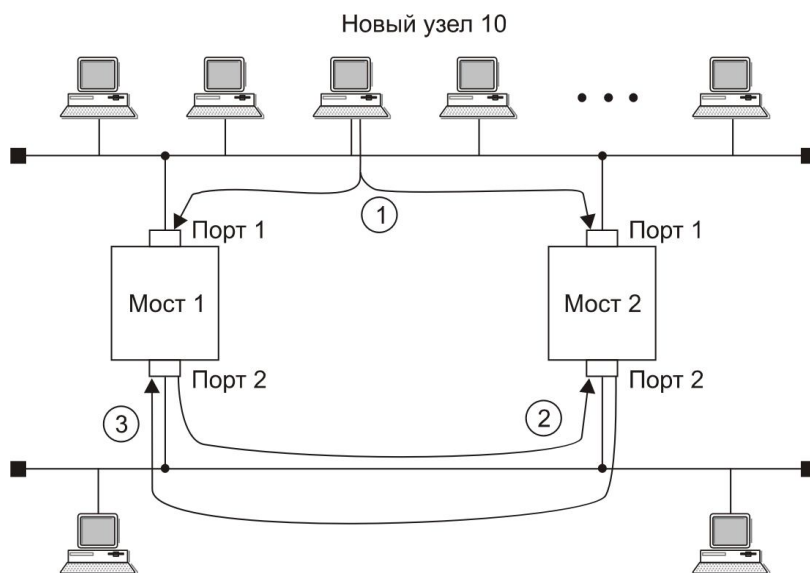


Рис. 4.16а. Влияние замкнутых маршрутов на работу коммутаторов

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. 4.16а). 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.

Последствия наличия петли в сети:

1. **«Размножение» кадра**, то есть появление нескольких его копий (двух в данном случае).
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.

Ограничения, связанные с применением мостов и коммутаторов, — **по топологии связей**, а также ряд других, — привели к тому, что в перечне коммуникационных устройств появилось еще одно устройство — **маршрутизатор (router)**. Маршрутизаторы более надежно и более эффективно, чем мосты, **изолируют трафик отдельных частей сети друг от друга**. Помимо локализации трафика маршрутизаторы выполняют еще много других полезных функций. Так, **маршрутизаторы могут работать в сети с замкнутыми контурами**, при этом

они обеспечивают выбор наиболее рациональных маршрутов. Другой важной функцией маршрутизаторов является их **способность связывать в единую сеть сети, построенные на базе разных сетевых технологий**, например Ethernet и ATM.

Помимо перечисленных устройств отдельные части сети может соединять **шлюз (gateway)**. Шлюз позволяет **объединять сети, построенные на существенно разных программных и аппаратных платформах**. Например, шлюз может позволить пользователям, работающим в сети **Unix**, взаимодействовать с пользователями сети **Windows**.

Традиционно в Интернете термины «шлюз» и «маршрутизатор» используются как синонимы.

Выводы

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

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

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

Разделяемой средой называется физическая среда передачи данных (коаксиальный кабель, витая пара, оптическое волокно, радиоволны), к которой непосредственно подключено несколько конечных узлов сети и которой они могут пользоваться только по очереди. В основе таких хорошо известных технологий, как Ethernet, FDDI, Token Ring, лежит принцип разделяемой среды. Хотя, казалось бы, сети на разделяемых средах уже пережили пик своей популярности, сегодня существуют явные признаки возрождения интереса к этой технологии, о чем свидетельствуют **домашние проводные сети, персональные и локальные беспроводные сети**. Во всех этих современных технологиях используется принцип разделения среды.

Вопросы и задания

1. Какие типы мультиплексирования и коммутации используются в телефонных сетях?
2. Какие свойства сетей с коммутацией каналов свидетельствуют об их недостатках?
3. Какие свойства сетей с коммутацией пакетов негативно сказываются на передаче мультимедийной информации?
4. Используется ли буферизация в сетях с коммутацией каналов?

5. Какой элемент сети с коммутацией каналов может отказать узлу в запросе на установление составного канала?
6. Какие концепции характерны для сетевой технологии Ethernet?
7. Учитывается ли в дейтаграммных сетях существование потоков данных?
8. Дайте определение логического соединения.
9. Можно ли организовать надежную передачу данных между двумя конечными узлами без установления логического соединения?
10. Какое логическое соединение может быть названо виртуальным каналом?
11. В каких сетях используется технология виртуальных каналов?
12. Укажите, какие из перечисленных устройств являются функционально подобными:
 - хаб;
 - коммутатор;
 - концентратор;
 - повторитель;
 - маршрутизатор;
 - мост.
13. Чем отличается мост от коммутатора?
14. Верно ли следующее утверждение: «Сеть Ethernet, имеющая звездообразную топологию с концентратором в центре, надежнее, чем та же сеть на коаксиальном кабеле, имеющая топологию общей шины»?
15. Как можно повысить пропускную способность, приходящуюся на компьютер каждого конечного пользователя, в сети, построенной на основе концентраторов?
16. Если все коммуникационные устройства в приведенном на рис. 4.17 фрагменте сети являются концентраторами, то на каких портах появится кадр, если его отправил компьютер А компьютеру В?
17. Если все коммуникационные устройства в приведенном на рис. 4.17 фрагменте сети являются коммутаторами, то на каких портах появится кадр, если его отправил компьютер А компьютеру В?
18. Если все коммуникационные устройства в приведенном на рис. 4.17 фрагменте сети являются коммутаторами, кроме одного концентратора, к которому подключены компьютеры А и В, то на каких портах появится кадр, если его отправил компьютер А компьютеру D?

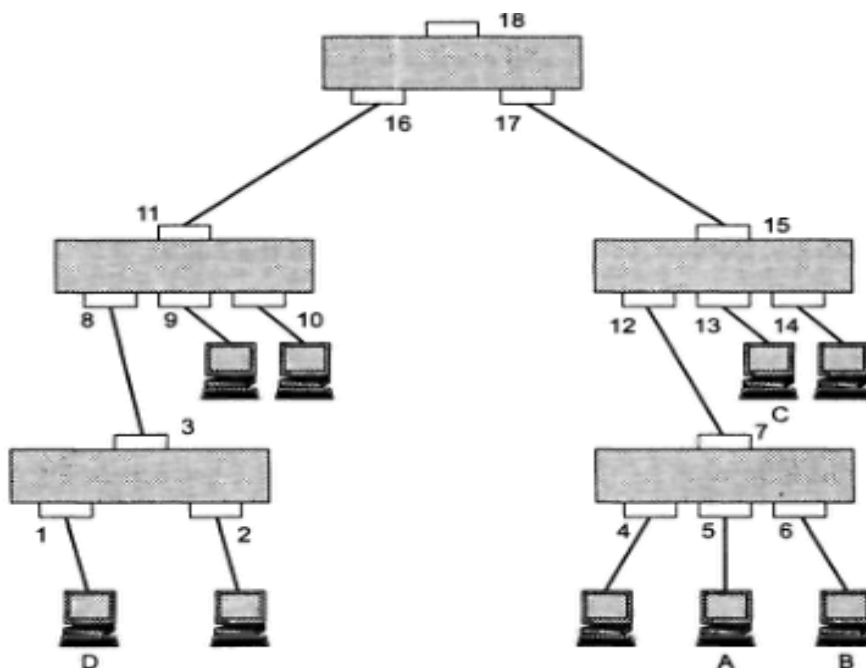


Рис. 3.22. Фрагмент сети

Рис. 4.17

19. В дейтаграммной сети между узлами А и В существует три потока и три альтернативных маршрута. Можно ли направить каждый поток по отдельному маршруту?

22. В сети, поддерживающей технику виртуальных каналов, между узлами А и В существует три потока и три альтернативных маршрута. Можно ли направить каждый поток по отдельному маршруту?

23. Сеть построена на разделяемой среде с пропускной способностью 10 Мбит/с и состоит из 100 узлов. С какой максимальной скоростью могут обмениваться данными два узла в сети?

24. Сеть может передавать данные в двух режимах: с помощью дейтаграмм и по виртуальным каналам. Какие соображения вы бы приняли во внимание при выборе того или иного режима для передачи ваших данных, если главным критерием выбора для вас является скорость и надежность доставки?

25. Считаете ли вы, что сети с коммутацией каналов в ближайшем будущем будут заменены сетями с коммутацией пакетов? Или же напротив, сети с коммутацией пакетов будут вытеснены сетями с коммутацией каналов? Или же эти технологии будут сосуществовать? Приведите аргументы в пользу вашего мнения. Рассмотрите разные области использования этих технологий.